
Video models are zero-shot learners and reasoners

Thaddäus Wiedemer^{*1} Yuxuan Li¹ Paul Vicol¹ Shixiang Shane Gu¹ Nick Matarese¹
Kevin Swersky¹ Been Kim¹ Priyank Jaini^{*1} Robert Geirhos^{*1}

Abstract

The remarkable zero-shot capabilities of Large Language Models (LLMs) have propelled natural language processing from task-specific models to unified, generalist foundation models. This transformation emerged from simple primitives: large, generative models trained on web-scale data. Curiously, the same primitives apply to today’s generative video models. Could video models be on a trajectory towards general-purpose *vision* understanding, much like LLMs developed general-purpose *language* understanding? We demonstrate that Veo 3 can solve a broad variety of tasks it wasn’t explicitly trained for: segmenting objects, detecting edges, editing images, understanding physical properties, recognizing object affordances, simulating tool use, and more. These abilities to perceive, model, and manipulate the visual world enable early forms of visual reasoning like maze and symmetry solving. Veo’s emergent zero-shot capabilities indicate that video models may be on a path to becoming unified, generalist vision foundation models. Project page with samples: <https://video-zero-shot.github.io>

1. Introduction

We believe that video models will become unifying, general-purpose foundation models for machine vision just like large language models (LLMs) have become foundation models for natural language processing (NLP). In the last few years, NLP underwent a radical transformation: from task-specific, bespoke models (e.g., one model for translation, another one for question-answering) to LLMs as unified foundation models. Here we ask: Do video mod-

¹Google DeepMind. Correspondence to: Thaddäus Wiedemer <thwiedemer@google.com>, Priyank Jaini <pjaini@google.com>, Robert Geirhos <last-name@google.com>.

els develop general-purpose *vision* understanding, similar to how LLMs developed general-purpose *language* understanding? We answer this question in the affirmative:

1. Analyzing 18,384 generated videos across 62 qualitative and 7 quantitative tasks, we report that Veo 3 can solve a wide range of tasks that it was neither trained nor adapted for.
2. Based on its ability to **perceive**, **model**, and **manipulate** the visual world, Veo 3 shows early forms of “chain-of-frames (CoF)” **visual reasoning** like maze and symmetry solving.
3. While task-specific bespoke models still outperform a zero-shot video model, we observe a substantial and consistent performance improvement from Veo 2 to Veo 3, indicating a rapid advancement in the capabilities of video models.

Tracking these emergent abilities (as we do via task-specific verifiers) enables measuring key building blocks that are essential for long-form video generation, such as the ability to maintain a persistent state from the first to the last frame, and the ability to translate complex prompts into a step-by-step visual execution.

2. Methods

Approach and motivation Our method is simple: We prompt Veo. This minimalist strategy is intentional, as it mirrors the transformation of NLP from task-specific fine-tuning or training to prompting a capable foundation model (Wei et al., 2022; Zhou et al., 2022; Liu et al., 2023). Here, we adopt the same philosophy to explore the capabilities of Veo 3 as a general-purpose vision model. Following Bommasani et al. (2021, p. 3), a foundation model is any model “trained on broad data (generally using self-supervision at scale)” which can be adapted (here: via prompting) “to a wide range of downstream tasks”.

Takeaway 1 In NLP, prompting replaced task-specific training or adaptation for many tasks. A similar paradigm shift is on the horizon in machine vision, facilitated by video models.

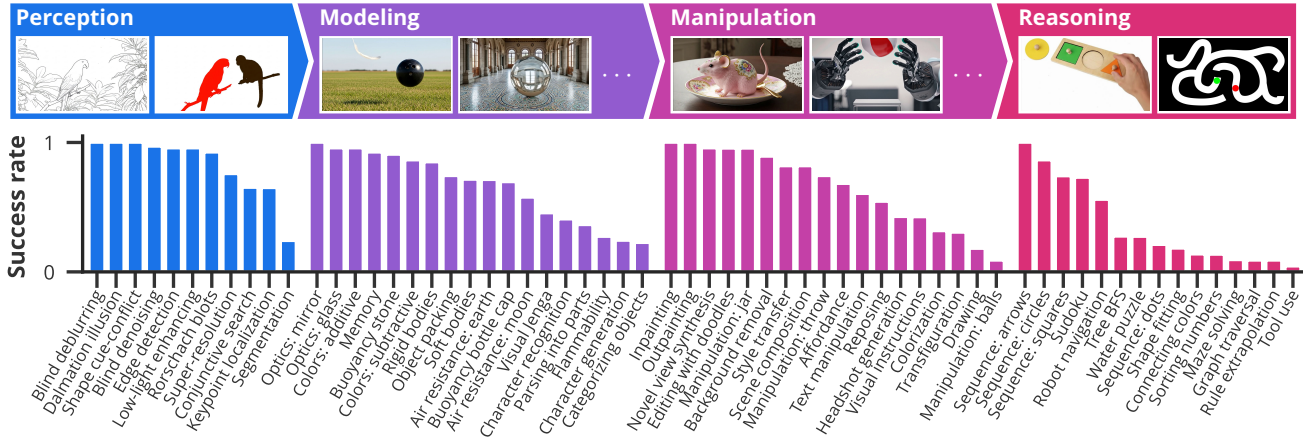


Figure 1. **Overview of Veo 3’s zero-shot abilities.** We show Veo 3’s success rate across 12 samples per task as a rough estimate of model performance on 62 tasks across the vision stack. Tasks are described in App. B. Find videos on our [project page](#).

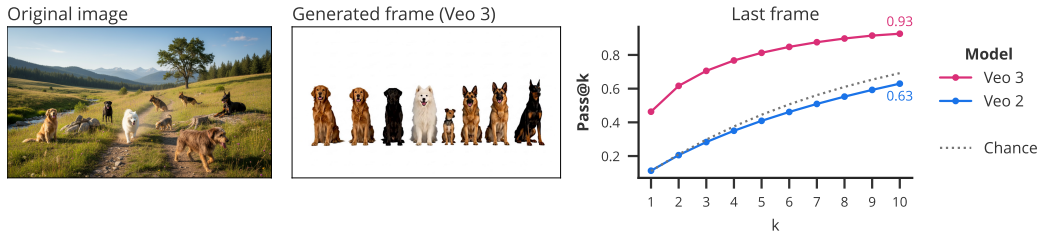


Figure 2. **Object extraction** on an animal dataset. Prompt: “The background changes to white [...] all animals line up in a row [...]” Details & full prompt: App. C.2.

Video generation For each task, we query the publicly available Veo 2 or Veo 3 models via Google Cloud’s Vertex AI API. We prompt the model with an initial input image (which the model uses as the first frame) and a text instruction. The models then generate a 16:9 video at 720p resolution, 24 FPS, for a duration of 8s. Veo 3 has model ID `veo-3.0-generate-preview` and Veo 2 model ID `veo-2.0-generate-001`.

Animated videos Videos can be viewed here: <https://video-zero-shot.github.io>

3. Results

We start by measuring Veo’s success rate across 62 different tasks through human raters in Fig. 1. The reader is encouraged to visit the [project page](#) to view the corresponding videos; see App. B for a detailed discussion. In a nutshell, Veo 3 is able to solve a broad variety of image2video tasks out-of-the box, without any task-specific training or adaptation—simply via prompting (including, for instance, low-light enhancement, material optics, novel view synthesis, or visual graph traversal). Motivated by these results, we then zoom in more closely to quantify performance of specific tasks through high-quality, task-specific automatic verifiers. We consider different facets of visual understanding: For **perception**, we assess Veo on edge detection. For

manipulation, we examine object extraction performance. Finally, we evaluate **reasoning** abilities through maze solving and visual symmetry solving.

We evaluate performance separately for the *best frame* and the *last frame* (where applicable). *Best frame* reports the best performance across any frame in the generated videos. This indicates the performance ceiling, but the optimal frame is not known a priori. Therefore, we also report performance on the *last frame* of each video, which may underestimate a model’s ability but has the practical advantage that the frame is predetermined. This distinction is important because Veo tends to continue animating a scene even after task completion, potentially reducing last frame performance. Where applicable, we use the state-of-the-art image editing model Nano Banana (Google, 2025) as a reference. As a general trend, we observe a large performance increase from Veo 2 to Veo 3, often matching or exceeding Nano Banana’s performance. Performance tends to improve substantially from $k = 1$ to $k = 10$ attempts, indicating that a good solution can be found in a reasonable number of tries. While image models are excellent zero-shot learners, too (Clark & Jains, 2023; Jains et al., 2023; Burgert et al., 2022; Khachatryan et al., 2023), video models are the more general framework because of their ability to process both temporal and spatial information.

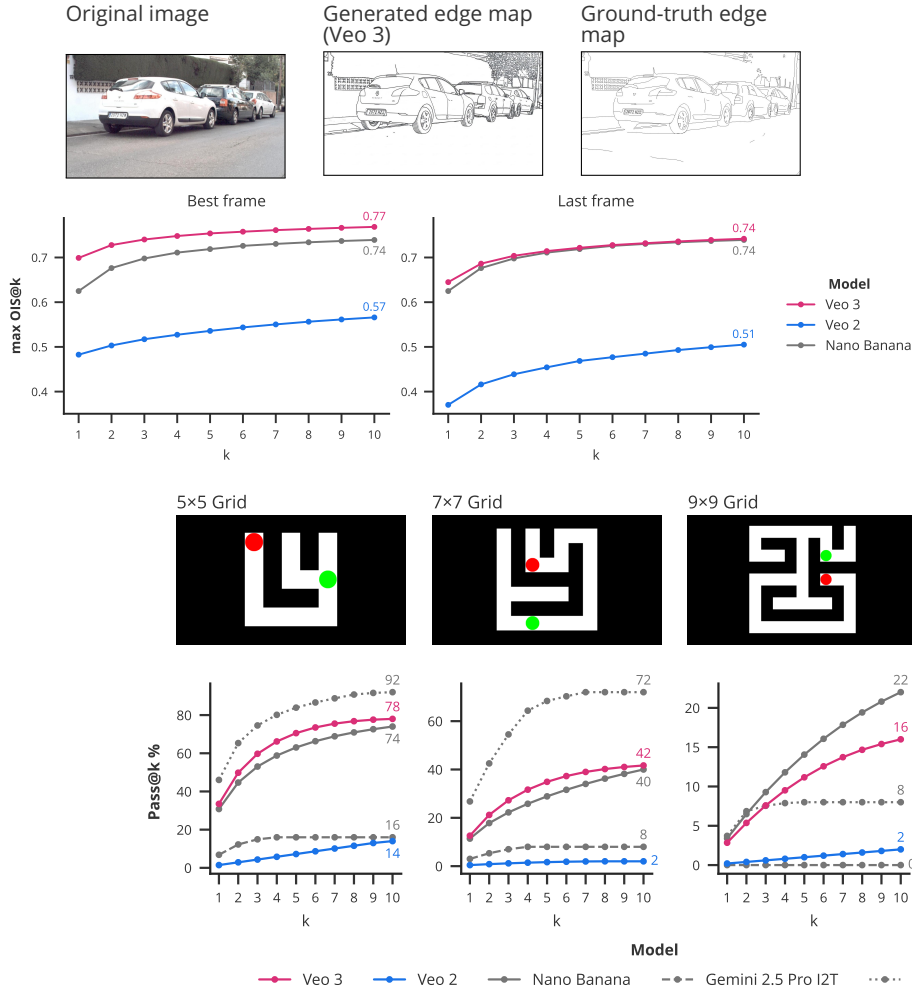


Figure 3. Edge detection on all 50 test images from BIPEDv2 (Soria et al., 2020; 2023). We generate 10 videos per sample and report best performance over k attempts as a function of k . Prompt: “All edges in this image become more salient by transforming into black outlines. Then, all objects fade away [...]” Details & full prompt: App. C.1.

Figure 4. Maze solving. Mazes of various sizes with start (red) and goal (green) locations. Prompt: “[...] The red circle slides smoothly along the white path, stopping perfectly on the green circle [...]” Details & full prompt: App. C.3. Veo 2 struggles to solve even small mazes, mostly due to early illegal moves. Veo 3 performs much better and benefits from multiple attempts. For comparison, we evaluate Nano Banana and also show Gemini 2.5 Pro’s performance on mazes presented as images (I2T) or ASCII text (T2T).

3.1. Perception: Edge detection

Despite not being trained for it, Veo 3 can be prompted to detect (i.e., **perceive**) edges. Fig. 3 details edge detection performance (measured by OIS; details and prompt in App. C.1) for Veo 2 and Veo 3. While Veo 3 (0.77 pass@10) lags behind task-specific SOTA (0.90) (Soria et al., 2023), its performance is remarkable for two reasons: First, it is zero-shot. Second, many of its edge maps are more detailed than the ground truth. For example, Veo 3 correctly outlines foliage and tire profiles; this hurts performance on the dataset but seems more indicative of a dataset limitation than a model limitation (the annotators understandably did not bother to trace each and every edge).

3.2. Manipulation: Object extraction

Can Veo perceive and extract (i.e., **manipulate**) all objects in a scene? We test this using a simple dataset depicting 1–9 animals (details in App. C.2). Veo is asked to extract and line up all animals horizontally, with white space between them (in some sense, a visual “tally”). To assess whether the number of extracted animals is correct, we count connected components in the last frame. Fig. 2 shows an example. While Veo 2 performs around chance, Veo 3 achieves 93% pass@10. This object extraction ability measures a capability that is required for long-form video generation: the ability to maintain a persistent state across different scenes.

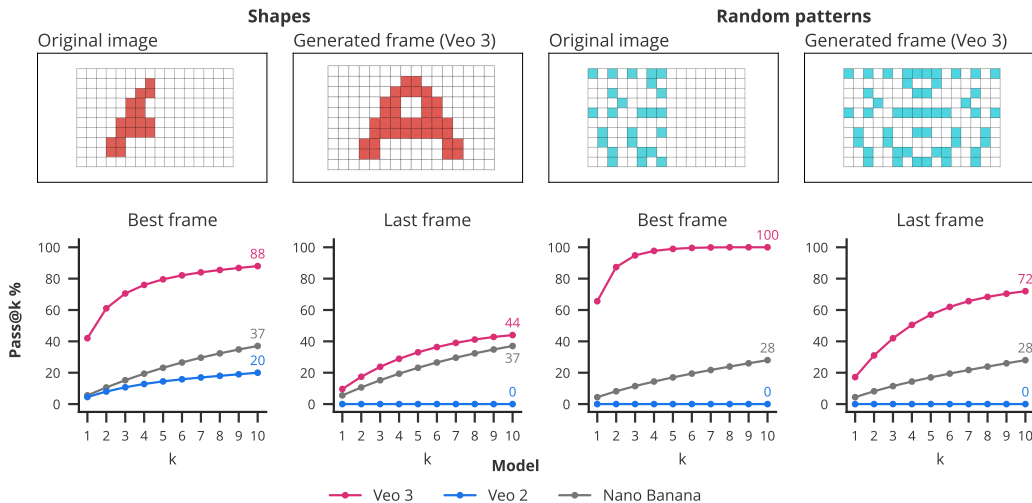


Figure 5. **Visual symmetry.** Prompt: “Instantly reflect this pattern along the central, vertical axis while keeping the existing colored pattern without modification. [...]” A model has to color all cells correctly to pass. Details & full prompt: App. C.4.

3.3. Reasoning: Maze solving

Maze solving tests a model’s ability to plan a path in a constrained environment, a key aspect of **reasoning**. In our setup, a red circle needs to navigate to a goal (green circle) without crossing any walls. We automatically verify the path (details in App. C.3) and present results for different mazes in Fig. 4. Veo 3 shows zero-shot maze solving abilities, significantly outperforming Veo 2 which often produces illegal moves. For instance, in 5×5 grids, Veo 3 achieves a pass@10 rate of 78% compared to Veo 2’s 14%. The consistent performance gap highlights the advancing reasoning capabilities of Veo 3. While Nano Banana matches or surpasses Veo 3’s performance on rectangular mazes, it fails to solve irregular mazes entirely. Similarly, Gemini 2.5 Pro outperforms Veo 3 on small mazes when given an ASCII representation of the maze (T2T), but falls behind on 9×9 mazes, and generally struggles when the maze is represented as image (as opposed to text) input. Both comparisons highlight the advantages of solving a visual task step-by-step in a visual medium.

Takeaway 2 Frame-by-frame video generation parallels chain-of-thought in language models. Just like chain-of-thought (CoT) enables language models to reason with symbols, a “chain-of-frames” (CoF) enables video models to reason across time and space.

3.4. Reasoning: Visual symmetry solving

Completing a pattern to be symmetrical assesses the ability to understand and apply spatial **reasoning**. We create a custom dataset of shapes (e.g., heart, letters) and random patterns (see App. C.4). Fig. 5 shows that Veo 3 outperforms Veo 2 and Nano Banana by a large margin. We also use this

task to systematically analyze how different prompts affect performance in App. D: The pass@1 difference between best and worst prompt is 40 percentage points on the shape split and 64 percentage points on the random split.

Takeaway 3 While not perfect, Veo 3—building on its ability to perceive, model and manipulate objects—shows emergent visual reasoning abilities.

4. Discussion

Summary A breakthrough in machine vision started the deep learning revolution in 2012 (Krizhevsky et al., 2012), but in recent years, NLP has seen the most rapid progress. This was driven by the rise of general-purpose LLMs, whose ability to solve tasks in a zero-shot fashion has led them to replace most task-specific models in NLP. We here make the case that machine vision is on the cusp of a similar paradigm shift, enabled by emergent abilities of large-scale video models. Our core finding is that Veo 3 can solve a wide range of tasks in a zero-shot manner, spanning the full vision stack from **perception** to **modeling**, **manipulation** and even early forms of **visual reasoning**. While performance is not perfect, the consistent improvement from Veo 2 to Veo 3 indicates that video models may become general-purpose foundation models for vision, just as LLMs have for language. We expect that a model’s visual abilities to perceive, model, manipulate and reason will co-evolve with the model’s abilities to perform long-form video generation. Progress on the underlying abilities such as prompt following, scene consistency, or object manipulation will directly translate not only into improved video generation for entertainment purposes, but also more generally into improved video foundation models.

Acknowledgements

We would like to thank Oyvind Taffjord, Mike Mozer, Katherine Hermann, Andrew Lampinen, Viorica Patraucean, Shiry Ginosar, Ross Goroshin, Abhijit Ogale, Claire Cui, Kun Zhang for helpful feedback/discussions, Anish Nangia for Vertex insights, Tuan Anh Le for suggesting the Omniglot evaluation, Medhini Narasimhan and Pieter-Jan Kindermans for Veo insights and discussion, Micromelon Robotics for permission to use the image in Fig. 55, Shelly Sheynin for permission to use the EMU Edit dataset, Fanfei Li for help with the human rating experiment, David Fleet, Jon Shlens and Noah Fiedel for their support, and the Veo team for developing an incredible model.

Author contributions

Leads: TW, PJ, RG. Project idea: RG. Core contributors: YL, PV. Partial contributor: SG. Fig. 35: NM. Advisors: KS, BK.

References

- Acuaviva, P., Davtyan, A., Hassan, M., Stapf, S., Rahimi, A., Alahi, A., and Favaro, P. Rethinking visual intelligence: Insights from video pretraining. *arXiv preprint arXiv:2510.24448*, 2025.
- Assran, M., Bardes, A., Fan, D., Garrido, Q., Howes, R., Muckley, M., Rizvi, A., Roberts, C., Sinha, K., Zholus, A., et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.
- Bansal, H., Lin, Z., Xie, T., Zong, Z., Yarom, M., Bitton, Y., Jiang, C., Sun, Y., Chang, K.-W., and Grover, A. Videophy: Evaluating physical commonsense for video generation, 2024.
- Bear, D. M., Wang, E., Mrowca, D., Binder, F. J., Tung, H.-Y. F., Pramod, R. T., Holdaway, C., Tao, S., Smith, K., Sun, F.-Y., Fei-Fei, L., Kanwisher, N., Tenenbaum, J. B., Yamins, D. L. K., and Fan, J. E. Physion: Evaluating physical prediction from vision in humans and machines, 2021.
- Bhattad, A., Preechakul, K., and Efros, A. A. Visual jenga: Discovering object dependencies via counterfactual inpainting. *arXiv preprint arXiv:2503.21770*, 2025.
- Bommasani, R., Hudson, D. A., Adeli, E., Altman, R. B., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N. S., Chen, A. S., Creel, K., Davis, J. Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L. E., Goel, K., Goodman, N. D., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D. E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P. W., Krass, M. S., Krishna, R., Kuditipudi, R., and et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.
- Burgert, R., Ranasinghe, K., Li, X., and Ryoo, M. S. Peekaboo: Text to image diffusion models are zero-shot segmentors. *arXiv preprint arXiv:2211.13224*, 2022.
- Campbell, D., Rane, S., Giallanza, T., De Sabbata, C. N., Ghods, K., Joshi, A., Ku, A., Frankland, S., Griffiths, T., Cohen, J. D., et al. Understanding the limits of vision language models through the lens of the binding problem. *Advances in Neural Information Processing Systems*, 37:113436–113460, 2024.
- Cherian, A., Corcodel, R., Jain, S., and Romeres, D. LLMPhy: Complex physical reasoning using large language models and world models. *arXiv preprint arXiv:2411.08027*, 2024.
- Chollet, F. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.
- Clark, K. and Jaini, P. Text-to-image diffusion models are zero shot classifiers. *Advances in Neural Information Processing Systems*, 36:58921–58937, 2023.
- Cohen, J. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.
- Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

- Cottier, B., Snodin, B., Owen, D., and Adamczewski, T. LLM inference prices have fallen rapidly but unequally across tasks, march 2025. URL <https://epoch.ai/data-insights/llm-inference-price-trends>. Accessed: 2025-09-12.
- Deitke, M., Liu, R., Wallingford, M., Ngo, H., Michel, O., Kusupati, A., Fan, A., Laforte, C., Voleti, V., Gadre, S. Y., et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36: 35799–35813, 2023.
- Dollár, P. and Zitnick, C. L. Structured forests for fast edge detection. In *ICCV*, 2013.
- Dollár, P. and Zitnick, C. L. Fast edge detection using structured forests. *ArXiv*, 2014.
- Fang, A., Ilharco, G., Wortsman, M., Wan, Y., Shankar, V., Dave, A., and Schmidt, L. Data determines distributional robustness in contrastive language image pre-training (clip). In *International conference on machine learning*, pp. 6216–6234. PMLR, 2022.
- Firestone, C. Performance vs. competence in human-machine comparisons. *Proceedings of the National Academy of Sciences*, 117(43):26562–26571, 2020.
- Garrido, Q., Ballas, N., Assran, M., Bardes, A., Najman, L., Rabbat, M., Dupoux, E., and LeCun, Y. Intuitive physics understanding emerges from self-supervised pretraining on natural videos. *arXiv preprint arXiv:2502.11831*, 2025.
- Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International conference on learning representations*, 2019.
- Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. Short-cut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020a.
- Geirhos, R., Meding, K., and Wichmann, F. A. Beyond accuracy: quantifying trial-by-trial behaviour of cnns and humans by measuring error consistency. *Advances in neural information processing systems*, 33:13890–13902, 2020b.
- Geng, D., Herrmann, C., Hur, J., Cole, F., Zhang, S., Pfaff, T., Lopez-Guevara, T., Doersch, C., Aytar, Y., Rubinstein, M., Sun, C., Wang, O., Owens, A., and Sun, D. Motion prompting: Controlling video generation with motion trajectories. *arXiv preprint arXiv:2412.02700*, 2024.
- Gillman, N., Herrmann, C., Freeman, M., Aggarwal, D., Luo, E., Sun, D., and Sun, C. Force prompting: Video generation models can learn and generalize physics-based control signals, 2025. URL <https://arxiv.org/abs/2505.19386>.
- Google. Veo 2 announcement. <https://blog.google/technology/google-labs/video-image-generation-update-december-2024/>, 2024. Accessed: September 22, 2025.
- Google. Nano Banana: Gemini Image Generation Overview. <https://gemini.google/overview/image-generation/>, 2025. Accessed: September 22, 2025.
- Google. Veo 2 launch. <https://developers.google.com/blog/en/veo-2-video-generation-now-generally-available/>, 2025a. Accessed: September 22, 2025.
- Google. Veo 3 announcement. <https://blog.google/technology/ai/generative-media-models-io-2025/>, 2025b. Accessed: September 22, 2025.
- Google. Veo 3 launch. <https://cloud.google.com/blog/products/ai-machine-learning/veo-3-fast-available-for-everyone-on-vertex-ai>, 2025c. Accessed: September 22, 2025.
- Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- Gregory, R. L. The intelligent eye, 1970.
- Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.
- Ivanitskiy, M. Maze dataset. <https://pypi.org/project/maze-dataset/0.3.4/>, 2025. Accessed: June 31, 2025.
- Jaech, A., Kalai, A., Lerer, A., Richardson, A., El-Kishky, A., Low, A., Helyar, A., Madry, A., Beutel, A., Carney, A., et al. OpenAI o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.
- Jaini, P., Clark, K., and Geirhos, R. Intriguing properties of generative classifiers. In *The Twelfth International Conference on Learning Representations*, 2023.
- James, R. C. Sight for sharp eyes. *LIFE*, 58(7):120, 1965.

- Jassim, S., Holubar, M., Richter, A., Wolff, C., Ohmer, X., and Bruni, E. GRASP: A novel benchmark for evaluating language grounding and situated physics understanding in multimodal language models. *arXiv preprint arXiv:2311.09048*, 2023.
- Jeong, Y., Uselis, A., Oh, S. J., and Rohrbach, A. Diffusion classifiers understand compositionality, but conditions apply. *arXiv preprint arXiv:2505.17955*, 2, 2025.
- Kang, B., Yue, Y., Lu, R., Lin, Z., Zhao, Y., Wang, K., Huang, G., and Feng, J. How far is video generation from world model: A physical law perspective. *arXiv preprint arXiv:2411.02385*, 2024.
- Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.
- Khachatryan, L., Movsisyan, A., Tadevosyan, V., Henschel, R., Wang, Z., Navasardyan, S., and Shi, H. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15954–15964, 2023.
- Khanam, R. and Hussain, M. Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*, 2024.
- Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.
- Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.
- Lampinen, A. K., Chan, S. C., Singh, A. K., and Shanahan, M. The broader spectrum of in-context learning. *arXiv preprint arXiv:2412.03782*, 2024.
- Leng, K., Zhang, Z., Liu, J., Boukhers, Z., Sui, W., Yang, C., and Li, Z. Superedge: Towards a generalization model for self-supervised edge detection. *CoRR*, 2024.
- Li, J., Fang, A., Smyrnis, G., Ivgi, M., Jordan, M., Gadre, S. Y., Bansal, H., Guha, E., Keh, S. S., Arora, K., et al. DataComp-LM: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282, 2024.
- Liu, D., Zhang, J., Dinh, A.-D., Park, E., Zhang, S., and Xu, C. Generative physical AI in vision: A survey. *arXiv preprint arXiv:2501.10928*, 2025.
- Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM computing surveys*, 55(9):1–35, 2023.
- LMSYS ORG. Lmsys org text-to-video leaderboard. <https://lmarena.ai/leaderboard/text-to-video>, September 2025. Accessed: 2025-09-23.
- Luo, Y., Zhao, X., Lin, B., Zhu, L., Tang, L., Liu, Y., Chen, Y.-C., Qian, S., Wang, X., and You, Y. V-ReasonBench: Toward unified reasoning benchmark suite for video generation models. *arXiv preprint arXiv:pdf/2511.16668*, 2025.
- Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegrefe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594, 2023.
- Mayilvahanan, P., Wiedemer, T., Mallick, S., Bethge, M., and Brendel, W. Llms on the line: Data determines loss-to-loss scaling laws, 2025a. URL <https://arxiv.org/abs/2502.12120>.
- Mayilvahanan, P., Zimmermann, R. S., Wiedemer, T., Rusak, E., Juhos, A., Bethge, M., and Brendel, W. In search of forgotten domain generalization, 2025b. URL <https://arxiv.org/abs/2410.08258>.
- McHugh, M. Interrater reliability: The kappa statistic. *Biochemia medica : časopis Hrvatskoga društva medicinskih biokemičara / HDMB*, 22:276–82, 10 2012. doi: 10.11613/BM.2012.031.
- Meng, F., Liao, J., Tan, X., Shao, W., Lu, Q., Zhang, K., Cheng, Y., Li, D., Qiao, Y., and Luo, P. Towards world simulator: Crafting physical commonsense-based benchmark for video generation, 2024.
- Motamed, S., Culp, L., Swersky, K., Jaini, P., and Geirhos, R. Do generative video models understand physical principles? *arXiv preprint arXiv:2501.09038*, 2025.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions

- with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- Peng, B., Li, C., He, P., Galley, M., and Gao, J. Instruction tuning with GPT-4. *arXiv preprint arXiv:2304.03277*, 2023.
- Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.
- Riochet, R., Castro, M. Y., Bernard, M., Lerer, A., Fergus, R., Izard, V., and Dupoux, E. IntPhys: A framework and benchmark for visual intuitive physics reasoning. *arXiv preprint arXiv:1803.07616*, 2018.
- Rotstein, N., Yona, G., Silver, D., Velich, R., Bensaid, D., and Kimmel, R. Pathways on the image manifold: Image editing via video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7857–7866, 2025.
- Sanli, E., Tezcan, B. S., Erdem, A., and Erdem, E. Can your model separate yolks with a water bottle? benchmarking physical commonsense understanding in video generation models. *arXiv preprint arXiv:2507.15824*, 2025.
- Schulze Buschoff, L. M., Akata, E., Bethge, M., and Schulz, E. Visual cognition in multimodal large language models. *Nature Machine Intelligence*, pp. 1–11, 2025.
- Snell, C., Lee, J., Xu, K., and Kumar, A. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.
- Soria, X., Riba, E., and Sappa, A. Dense extreme inception network: Towards a robust CNN model for edge detection. In *The IEEE Winter Conference on Applications of Computer Vision (WACV '20)*, 2020.
- Soria, X., Sappa, A., Humanante, P., and Akbarinia, A. Dense extreme inception network for edge detection. *Pattern Recognition*, 139:109461, 2023. ISSN 0031-3203. doi: <https://doi.org/10.1016/j.patcog.2023.109461>. URL <https://www.sciencedirect.com/science/article/pii/S0031320323001619>.
- Tong, J., Mou, Y., Li, H., Li, M., Yang, Y., Zhang, M., Chen, Q., Liang, T., Hu, X., Zheng, Y., et al. Thinking with video: Video generation as a promising multimodal reasoning paradigm. *arXiv preprint arXiv:2511.04570*, 2025.
- Tung, H.-Y., Ding, M., Chen, Z., Bear, D., Gan, C., Tenenbaum, J., Yamins, D., Fan, J., and Smith, K. Physion++: Evaluating physical scene understanding that requires online inference of different physical properties. *Advances in Neural Information Processing Systems*, 36, 2024.
- Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837, 2022.
- Weihs, L., Yuile, A., Baillargeon, R., Fisher, C., Marcus, G., Mottaghi, R., and Kembhavi, A. Benchmarking progress to infant-level physical reasoning in ai. *Transactions on Machine Learning Research*, 2022.
- Wiedemer, T., Sharma, Y., Prabhu, A., Bethge, M., and Brendel, W. Pretraining frequency predicts compositional generalization of clip on real-world tasks, 2025. URL <https://arxiv.org/abs/2502.18326>.
- Wu, W., Mao, S., Zhang, Y., Xia, Y., Dong, L., Cui, L., and Wei, F. Mind’s eye of LLMs: visualization-of-thought elicits spatial reasoning in large language models. *Advances in Neural Information Processing Systems*, 37: 90277–90317, 2024.
- Xie, S. and Tu, Z. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 1395–1403, 2015.
- Yang, S., Walker, J., Parker-Holder, J., Du, Y., Bruce, J., Barreto, A., Abbeel, P., and Schuurmans, D. Video as the new language for real-world decision making. *arXiv preprint arXiv:2402.17139*, 2024.
- Yang, W., Wang, W., Huang, H., Wang, S., and Liu, J. Sparse gradient regularized deep retinex network for robust low-light image enhancement. *IEEE Transactions on Image Processing*, 30:2072–2086, 2021.
- Zamir, A. R., Sax, A., Shen, W., Guibas, L. J., Malik, J., and Savarese, S. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3712–3722, 2018.

Zhang, C., Cherniavskii, D., Zadaianchuk, A., Tragoudaras, A., Vozikis, A., Nijdam, T., Prinzhorn, D. W., Bodracska, M., Sebe, N., and Gavves, E. Morpheus: Benchmarking physical reasoning of video generative models with real physical experiments. *arXiv preprint arXiv:2504.02918*, 2025.

Zhou, Y., Muresanu, A. I., Han, Z., Paster, K., Pitis, S., Chan, H., and Ba, J. Large language models are human-level prompt engineers. In *The eleventh international conference on learning representations*, 2022.

Zitnick, C. L. and Dollár, P. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014.

Appendix

Animated samples can be accessed here: <https://video-zero-shot.github.io>

A. Extended discussion

Zero-shot learning in the era of web-scale data The advent of models trained on web-scale data has led to a necessary evolution in the understanding of zero-shot capabilities. Given the vast and diverse nature of the training data, it is difficult to verify that a model has not encountered data similar to a particular task before. In contemporary usage, the definition of “zero-shot learning” has therefore shifted from a strict “unseen task” criterion (often impossible to verify) to a methodological one which we adopt: A task is considered zero-shot if the model can be prompted to perform the task despite not being trained, adapted or fine-tuned specifically for that task (Kojima et al., 2022; Liu et al., 2023; Bubeck et al., 2023). For instance, LLMs were described as “zero-shot reasoners” (Kojima et al., 2022) not because of any guarantees that reasoning-style text isn’t part of the training data by accident, but because they can perform reasoning tasks without task-specific training or adaptation, simply via prompting alone.

Performance is a lower bound Tasks can be represented in a myriad of ways; a maze, for example, can be presented as a black-and-white grid, a video game, or a photorealistic scene, with the prompt requesting a solution in the form of a line, a moving object, or a glowing path. Moreover, visually, a maze could be represented as a black-and-white grid, a Pac-Man game, or a photorealistic top-down view of an apartment. This has three implications: First, prompt engineering—including the *visual* prompt a.k.a. starting frame—is as important for visual tasks as it is for LLMs (see also App. D and Lampinen et al. (2024) for a discussion). Second, we must distinguish between a model’s task performance and its underlying ability (i.e., competence) to solve that task (Firestone, 2020; Geirhos et al., 2020a). Third, as a consequence, the model performance reported here with a given visual and textual prompt should be considered a lower bound on the model’s true capabilities. This also holds for the tasks that we report as failure cases in App. F, such as providing visual instructions to fold laundry (Fig. 72), planning to move a sofa between rooms (Fig. 73), or certain visual puzzles (Fig. 66).

Video generation is expensive, but costs tend to fall Generating a video is more expensive than running a task-specific model (see also App. E). Yet, the economics of general-purpose models are on a predictable trajectory: Epoch AI (Cottier et al., 2025) estimates that LLM inference costs are falling by a factor of $9\times$ to $900\times$ per year for a given performance level. In NLP, early generalist models were also considered prohibitively expensive “GPT-3’s size makes it challenging to deploy” (“GPT-3’s size makes it challenging to deploy”, Brown et al., 2020, p. 8). Nevertheless, rapidly falling inference costs, combined with the appeal of generalist models, have replaced most task-specific language models. If NLP is a guide, the same trend will play out in vision. While specialized models will likely remain cheaper, the superior performance and cross-task synergies of foundation models justify their higher cost. For example, today’s leading coding models aren’t task-specific models, but general-purpose LLMs. Similarly, unified vision models will likely outperform many task-specific models since visual tasks have deep relationships with each other (Zamir et al., 2018), enabling generalist models to benefit from synergies.

Jack of many trades, master of few? For many tasks, Veo 3’s performance is below state of the art of specialized models. This mirrors the early days of LLMs; GPT-3 reported performance well below fine-tuned models on many tasks (Brown et al., 2020, cf. Tables 3.1, 3.3, 3.4, 3.5). This did not stop LLMs from becoming foundation models, and we don’t believe it will stop video models from becoming vision foundation models for two reasons. First, the step-change in performance from Veo 2 to Veo 3 is evidence of rapid progress. Second, our scaling results from Sec. 3 show $\text{pass}@10$ to be consistently higher than $\text{pass}@1$ with no signs of a plateau. Therefore, inference-time scaling methods (e.g. Wang et al., 2022; Madaan et al., 2023; Jaech et al., 2024; Snell et al., 2024) in combination with the standard optimization toolkit like post-training with automatic verifiers are likely to boost performance. For the tasks we test here, Veo 3 is akin to a pre-trained language model that has yet to undergo instruction tuning or RLHF (Ouyang et al., 2022; Peng et al., 2023).

Why Veo? The core argument of this paper—that video models are zero-shot learners and reasoners—can be supported by demonstrating success on *any* sufficiently capable model. We choose Veo for its consistent high ranking on `text2video` and `image2video` leaderboards (LMSYS ORG, 2025). Unless noted otherwise, figures are generated with Veo 3. To provide a sense of how rapidly performance is improving, our quantitative analyses compare Veo 3 with its predecessor, Veo 2, released roughly half a year earlier: Veo 2 was announced in December 2024 and released in April 2025 (Google, 2024; 2025a), while Veo 3 was announced in May 2025 and released in July 2025 (Google, 2025b;c).

Will other video models show similar behavior? Articles released after the initial version of this work confirm that other

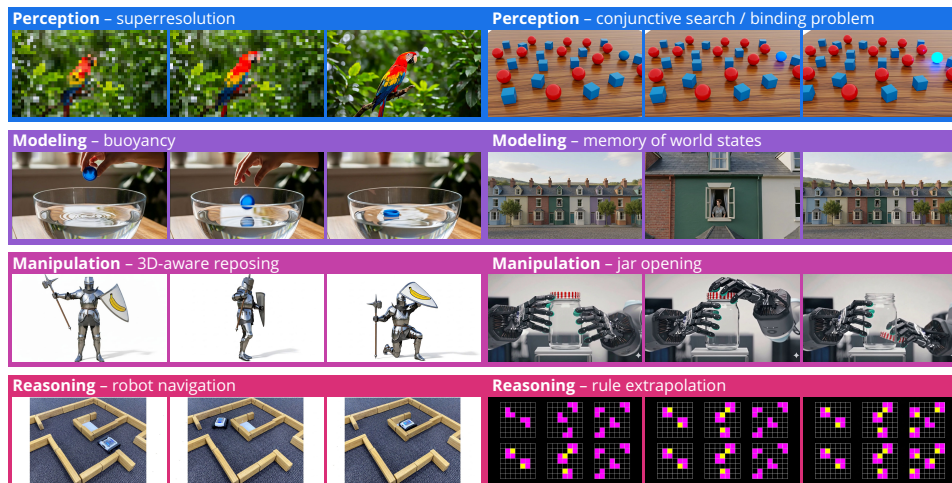


Figure 6. Veo 3 zero-shot learning and reasoning examples. From classic **perceptual** tasks (superresolution, visual search) to **modeling** (buoyancy, memory of world states after zooming in), **manipulation** (pose editing, simulating dexterous manipulation) and **visual reasoning** (navigation, rule extrapolation): Veo 3 can zero-shot solve a host of visual tasks that are specified as an input image and a text prompt. Examples are shown in App. B; videos of all tasks are on our [project page](#).

models also show strong performance: [Tong et al. \(2025\)](#) investigate Sora-2, [Luo et al. \(2025\)](#) test six models including Sora-2, Hailuo-02 and Kling-2.5-Turbo-Pro, while [Acuaviva et al. \(2025\)](#) adapt CogVideoX1.5-5B for visual reasoning tasks. Collectively, they indicate that emergent visual intelligence in video models isn’t limited to a single model.

Where does Veo’s performance come from? Veo’s architecture, training data mixture, and training recipe, alongside the exact models used for embeddings and prompt-rewriting, and the specific verbal and visual prompts used in each task all likely influence the performance reported in this work. Related works published after the initial version of this manuscript and mentioned in the paragraph above indicate that many of the behaviors we demonstrate are not specific to Veo. Instead, they seem to be an emergent property of video generation trained on large-scale data. As is the case for large language and vision models ([Fang et al., 2022](#); [Li et al., 2024](#); [Mayilvahanan et al., 2025a;b](#); [Wiedemer et al., 2025](#)), the exact performance distribution across tasks is likely determined by the data mixture, though more research is required. It remains to be seen whether scaling laws can be established for visual intelligence tasks of video models, capturing the impact of data scale and compute similarly to prior works in LLMs ([Kaplan et al., 2020](#); [Hoffmann et al., 2022](#); [Grattafiori et al., 2024](#)). See App. G for a discussion on similarities and differences to NLP.

B. Qualitative results: Sparks of visual intelligence?

We begin with a comprehensive, qualitative investigation across visual tasks to assess the potential of video models as visual foundation models. We organize our findings into four hierarchical capabilities, each building on the preceding ones (c.f. Fig. 1 and Fig. 6):

1. **Perception** as a foundational ability to understand visual information.
2. **Modeling**, which builds on the perception of objects to form a model of the visual world.
3. **Manipulation**, which meaningfully alters the perceived and modeled world.
4. **Reasoning** across space and time over a sequence of manipulation steps.

While capability boundaries often overlap, this hierarchy provides a framework for understanding the emergent abilities of video models. For example, solving a maze (see Sec. 3.3) requires perceiving the maze, modeling its state (walls vs. floor), and finally manipulating an object (a mouse, a circle) to move from start to finish. For each task in this section, we prompt Veo 3 22 times and report the *success rate* in the caption. *Success rate* is the fraction of videos that solve a task, as determined by human annotators (see App. B.5). A success rate greater than 0 suggests that the model possesses the *ability* to solve the task, while a success rate closer to 1 indicates that the task is solved *reliably* irrespective of the random seed. While not a substitute for the systematic quantification we perform in Sec. 3, this provides a ballpark estimate of the model’s capabilities.

Perception Computer vision has historically relied on a suite of specialized models for tasks like segmentation ([Kirillov et al., 2023](#); [Ravi et al., 2024](#)), object detection ([Redmon et al., 2016](#); [Khanam & Hussain, 2024](#)), and edge detection ([Xie & Tu, 2015](#)). While some backbones can be adapted or fine-tuned for other tasks, training-free transfer to novel tasks is rare, limiting generalization. As we show here, this is changing with large video models. Without any task-specific training,

Veo 3 can perform a range of classic computer vision tasks, including edge detection (Fig. 7), segmentation (Fig. 8), keypoint localization (Fig. 9), super-resolution (Fig. 10), blind deblurring (Fig. 11), denoising (Fig. 12) and low-light enhancing (Fig. 13). Some of these tasks were already demonstrated in Rotstein et al. (2025), but Veo’s emergent abilities extend farther: It can perform complex tasks like conjunctive search (Fig. 14) and interpreting ambiguous images such as the classic dalmatian illusion (Fig. 15), the cat shape in a texture-shape cue conflict image (Fig. 16), and colored blots from the Rorschach test (Fig. 17). Apart from denoising—the classic diffusion objective—none of these tasks are explicitly trained for in video models.

Takeaway 4 Veo 3 shows emergent zero-shot perceptual abilities well beyond the training task. Just like LLMs replaced many task-specific NLP models, we believe that generative vision models will replace most bespoke models in computer vision—once they become sufficiently cheap and reliable.

Modeling: intuitive physics & world models Based on their *perception* of the visual world, video models are starting to *model* it, too. Modeling the world and the principles that govern it (e.g., laws of physics) is a critical step toward successful prediction and action. Several works have investigated and quantified intuitive physics in deep models (e.g., Riochet et al., 2018; Bear et al., 2021; Weihs et al., 2022; Jassim et al., 2023; Tung et al., 2024; Bansal et al., 2024; Cherian et al., 2024; Meng et al., 2024; Kang et al., 2024; Motamed et al., 2025; Liu et al., 2025; Sanli et al., 2025; Schulze Buschoff et al., 2025; Zhang et al., 2025; Garrido et al., 2025). Here, we investigate an exemplary subset of tasks from these works. Veo’s grasp of physics is demonstrated by its ability to model various physical properties, like flammability (Fig. 18), rigid and soft body dynamics and their surface interactions (Fig. 19) and air resistance affecting falling objects (Fig. 20), and buoyancy (Fig. 21). As illustrated by the Visual Jenga task (Bhattad et al., 2025), Veo can remove objects from a scene in a physically plausible order (Fig. 22) and understands which objects fit into a backpack (Fig. 23). Furthermore, it correctly generates some optical phenomena like refraction and reflection (Fig. 24) and additive/subtractive color mixing (Fig. 25). Beyond physical characteristics, Veo understands some abstract relationships which is an important aspect of modeling the world. For example, Veo can distinguish categories like toys from other objects like a laptop (Fig. 26). On samples inspired by the Omniglot dataset (Lake et al., 2015), we demonstrate Veo’s ability to recognize patterns, generate variations thereof, and parse larger wholes into parts (Fig. 27). Lastly, Veo maintains a memory of the world state across time and camera movements within the video context (Fig. 28).

Manipulation: editing & imagination Based on its ability to *perceive* objects and *model* their relation to each other and the world, Veo can meaningfully *manipulate* the visual world, too. This enables Veo 3 to perform a variety of zero-shot image editing tasks like background removal (Fig. 29), style transfer (Fig. 30), colorization (Fig. 31), inpainting (Fig. 32), and outpainting (Fig. 33). Furthermore, it can manipulate text elements (Fig. 34), and edit images based on doodle instructions (Fig. 35). Veo’s understanding of 3D world enables it to compose scenes from individual components (Fig. 36), generate novel views of objects and characters (Figs. 37 and 38), smoothly transform one object into another (Fig. 39), or change the perspective, lighting, and appearance to turn a selfie into a professional photograph (Fig. 40). This ability to plausibly modify a scene allows it to imagine complex interactions, simulate dexterous object manipulation (Fig. 41; note that we do not test actual robots as e.g. Assran et al. (2025) do), interpreting object affordances (Fig. 42), demonstrating how to draw a shape (Fig. 43) and roll a burrito (Fig. 44). Overall, video models can meaningfully manipulate and simulate aspects of the (digital) visual world.

Visual reasoning across time and space *Perception, modeling, and manipulation* all integrate to tackle *visual reasoning*. While language models manipulate human-invented symbols, video models can apply changes across the dimensions of the real world: time and space. Since these changes are applied frame-by-frame in a generated video, this parallels chain-of-thought in LLMs and could therefore be called *chain-of-frames*, or CoF for short. In the language domain, chain-of-thought enabled models to tackle reasoning problems (Wei et al., 2022), and visualizing intermediate steps helps (Wu et al., 2024). Similarly, chain-of-frames (a.k.a. video generation) might enable video models to solve challenging visual problems that require step-by-step reasoning across time and space. We see early signs of this ability in tasks such as generating a valid graph traversal (Fig. 45), performing visual breadth-first search on a tree (Fig. 46), completing visual sequences (Fig. 47), connecting matching colors (Fig. 48), fitting shapes into holes (Fig. 49), and sorting numbers (Fig. 50). Furthermore, Veo can use tools to accomplish a visual task (Fig. 51) and solve simple Sudokus (Fig. 52) or visual puzzles (Fig. 53). Finally, it can solve mazes and navigation tasks (Figs. 54 and 55 and Sec. 3.3) and extrapolate rules from visual examples (Fig. 56). While not always perfect, the model’s ability to solve such problems in a zero-shot manner points to exciting possibilities for more advanced visual reasoning and planning in future, with more capable video models.

Summary Taken together, the qualitative examples from this section indicate that a capable video model like Veo 3 possesses strong zero-shot learning abilities. While the results are not always perfect, the model consistently demonstrates the capacity to solve a wide variety of tasks for which it was not explicitly trained.

We show the initial prompt frame and successive generated frames for each task described in App. B. For each task, the figure caption also states the prompt used to generate all samples. The guidelines used to determine success in each task are listed in App. B.5.

B.1. Perception

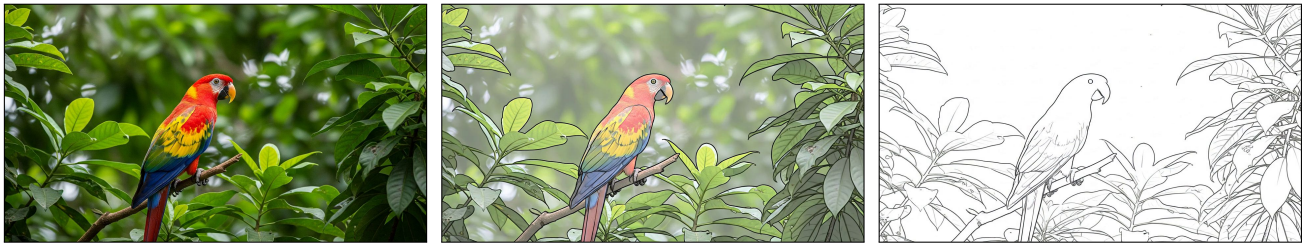


Figure 7. **Edge detection.** Prompt: “All edges in this image become more salient by transforming into black outlines. Then, all objects fade away, with just the edges remaining on a white background. Static camera perspective, no zoom or pan.” Success rate: 0.96.

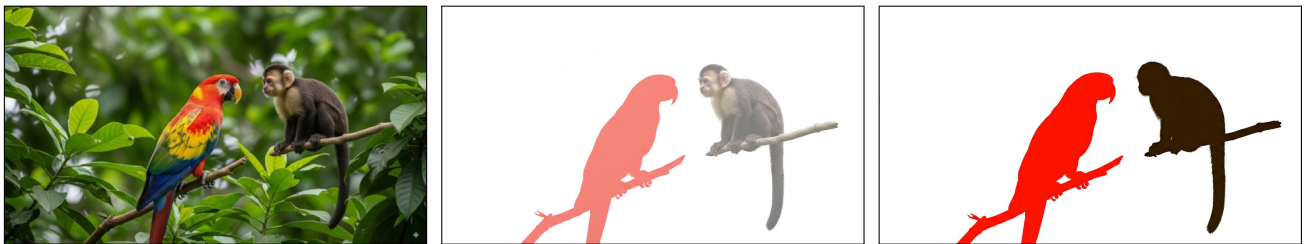


Figure 8. **Segmentation.** Prompt: “Create an animation of instance segmentation being performed on this photograph: each distinct entity is overlaid in a different flat color [...]” (full prompt: ??). Success rate: 0.24.

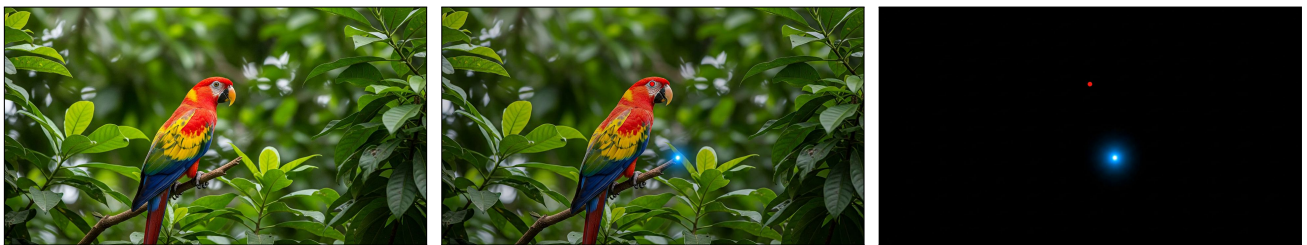


Figure 9. **Keypoint localization.** Prompt: “Add a bright blue dot at the tip of the branch on which the macaw is sitting. The macaw’s eye turns bright red. Everything else turns pitch black. Static camera perspective, no zoom or pan.” Success rate: 0.65.

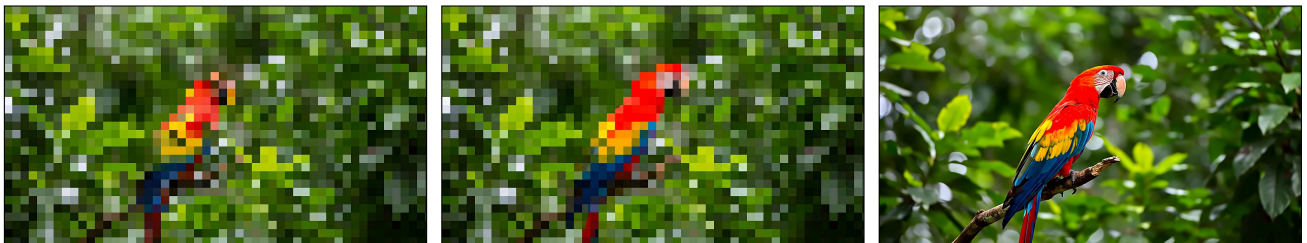


Figure 10. **Super-resolution.** Prompt: “Perform superresolution on this image. Static camera perspective, no zoom or pan.” Success rate: 0.76.

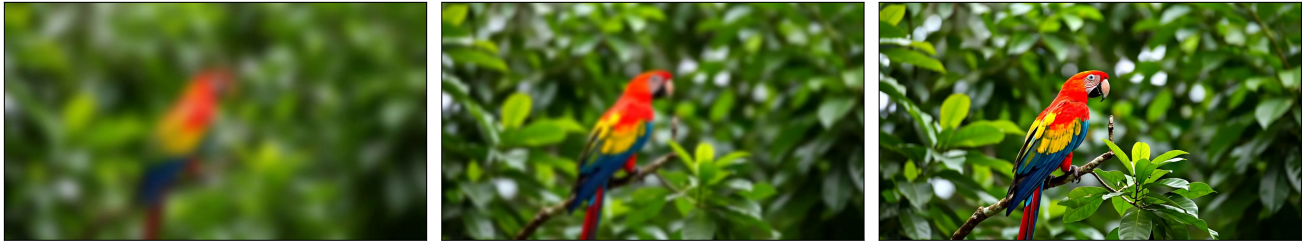


Figure 11. **Blind deblurring.** Prompt: “Unblur image including background. Static camera perspective, no zoom or pan.” Success rate: 1.0.

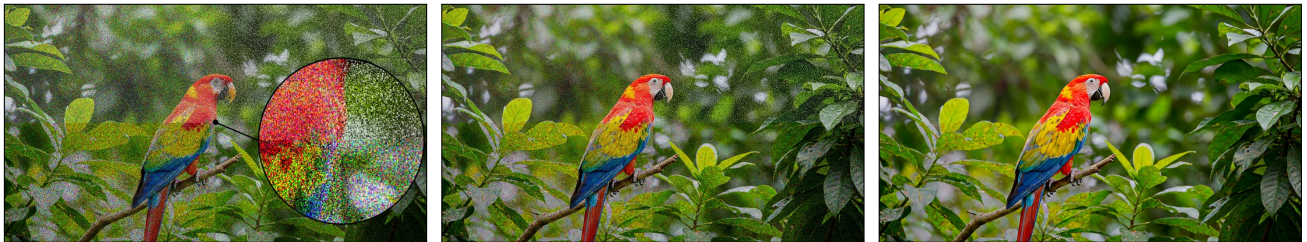
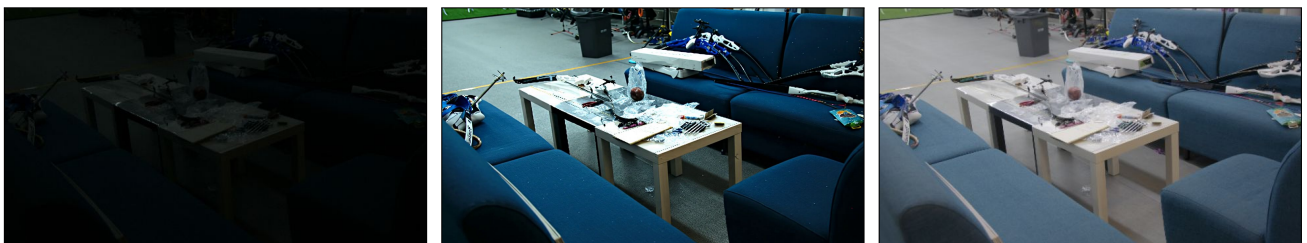


Figure 12. **Blind denoising.** Each quadrant was corrupted with a different type of noise. Clockwise from top left: Gaussian noise, salt-and-pepper noise, speckle noise, shot noise. Prompt: “Remove the noise from this image. Static camera perspective, no zoom or pan.” Success rate: 0.97.



Original low-light image

Veo 3-generated lit image

Ground-truth lit image

Figure 13. **Low-light enhancing.** Prompt: “Fully restore the light in this image. Static camera perspective, no zoom or pan.” Success rate: 0.96. Image and ground-truth source: LOLv2 dataset (Yang et al., 2021).

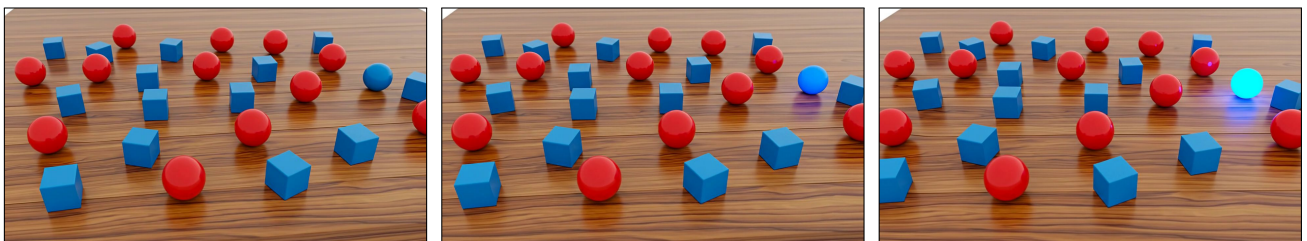


Figure 14. **Conjunctive search / binding problem.** Prompt: “The blue ball instantly begins to glow. Static camera perspective, no zoom no pan no movement no dolly no rotation.” Success rate: 0.65. Inspiration: (Campbell et al., 2024).



Figure 15. **Dalmatian illusion understanding.** Prompt: “Static camera perspective.” Success rate: 1.0. Image credit: (James, 1965; Gregory, 1970).

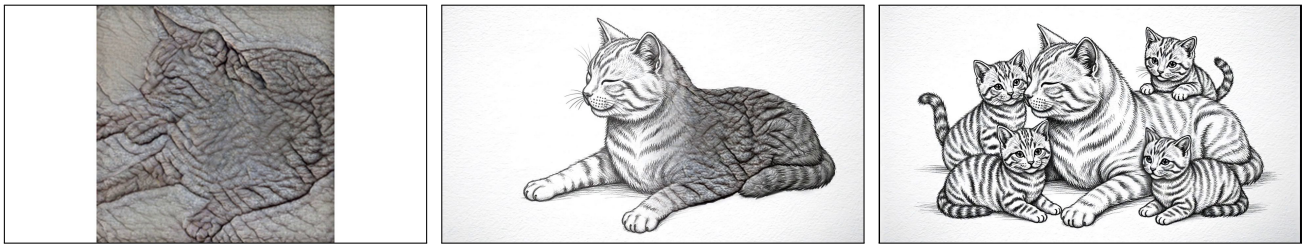


Figure 16. **Shape (cue-conflict) understanding.** Prompt: “Transform the animal in this image into a sketch of the animal surrounded by its family.” Success rate: 1.0. Image credit: (Geirhos et al., 2019).

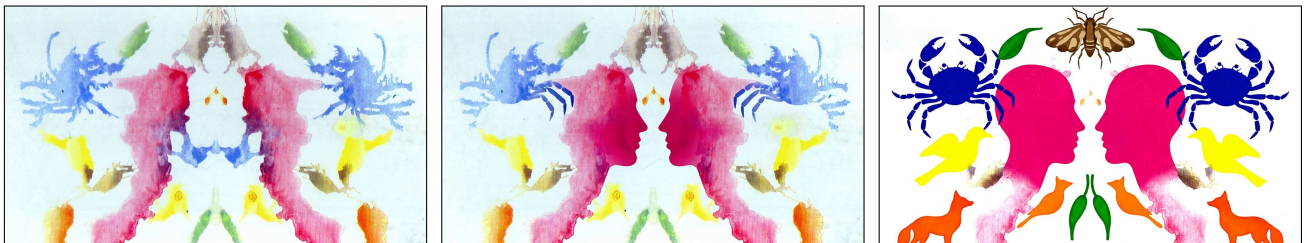


Figure 17. **Rorschach blot interpretation.** Prompt: “The patterns transform into objects.” Success rate: undefined (0.92 for prompt following). Image credit: H. Rorschach, public domain via [wikipedia](https://en.wikipedia.org/wiki/Hans_Rorschach).

B.2. Modeling



Figure 18. **Material properties.** Prompt: “The bunsen burner at the bottom turns on. Sped up time lapse. Static camera, no pan, no zoom, no dolly.” Success rate: 0.27.

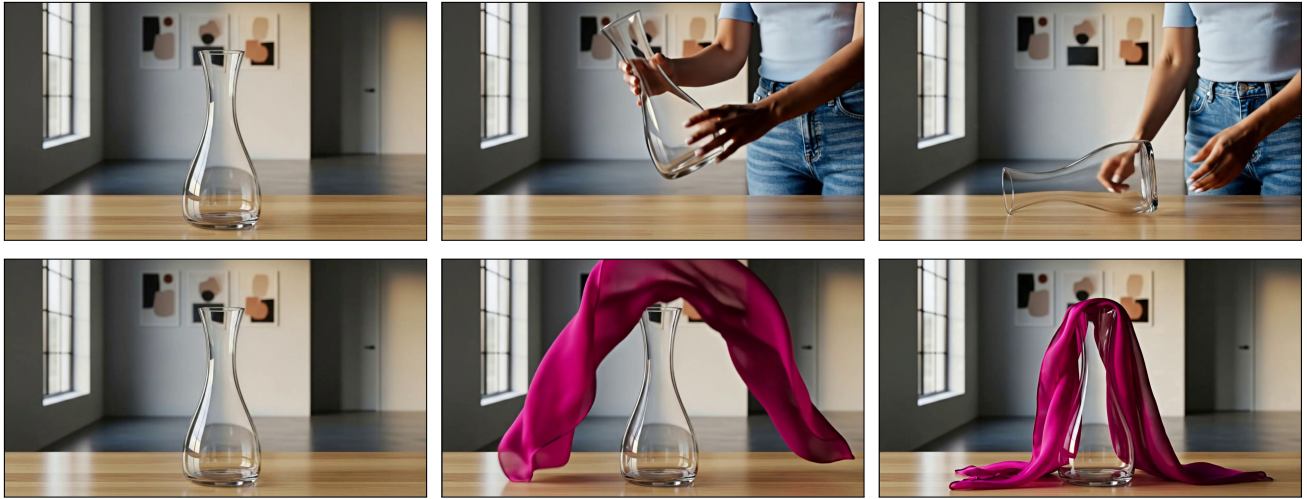


Figure 19. **Physics body transform. Rigid body** (top). Prompt: “A person picks up the vase and puts it back on the table in a sideways orientation. Static camera, no pan, no zoom, no dolly.” Success rate: 0.85. **Soft body** (bottom). Prompt: “A person drapes a thin silk scarf over the vase. Static camera, no pan, no zoom, no dolly.” Success rate: 0.71.

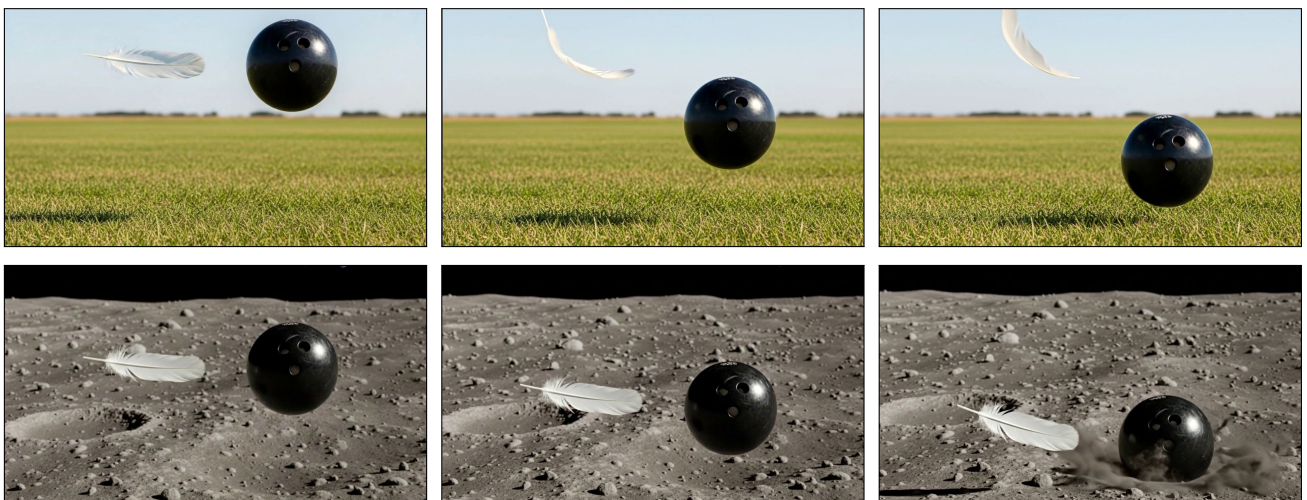


Figure 20. **Gravity and air resistance. On earth** (top). Prompt: “The objects fall due to gravity. Static camera, no pan, no zoom, no dolly.” Success rate: 0.71. **On the moon** (bottom). Prompt: “The objects fall down on the moon due to gravity. Static camera, no pan, no zoom, no dolly.” Success rate: 0.58.

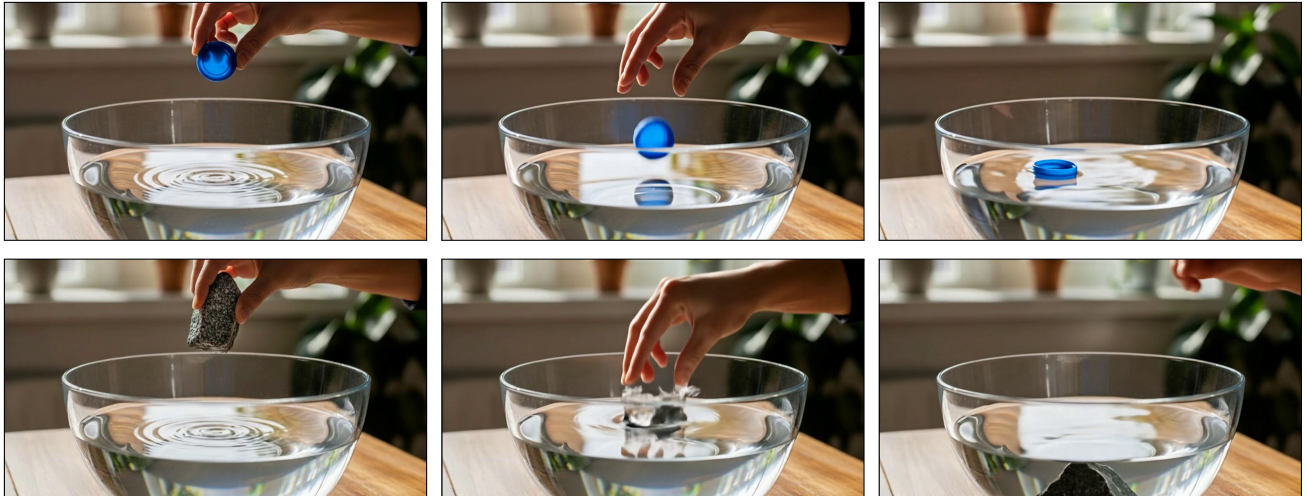


Figure 21. **Buoyancy**. Prompt: “The hand lets go of the object. Static camera, no pan, no zoom, no dolly.” Success rate (bottle cap): 0.70; success rate (rock): 0.91.



Figure 22. **Visual Jenga**, inspired by (Bhattad et al., 2025). Prompt: “A hand quickly removes each of the items in this image, one at a time.” Success rate, based on removal of at least three objects: 0.45.



Figure 23. **Object packing**. Prompt: “A person puts all the objects that can fit in the backpack inside of it. Static camera, no pan, no zoom, no dolly.” Success rate: 0.74.

Video models are zero-shot learners and reasoners

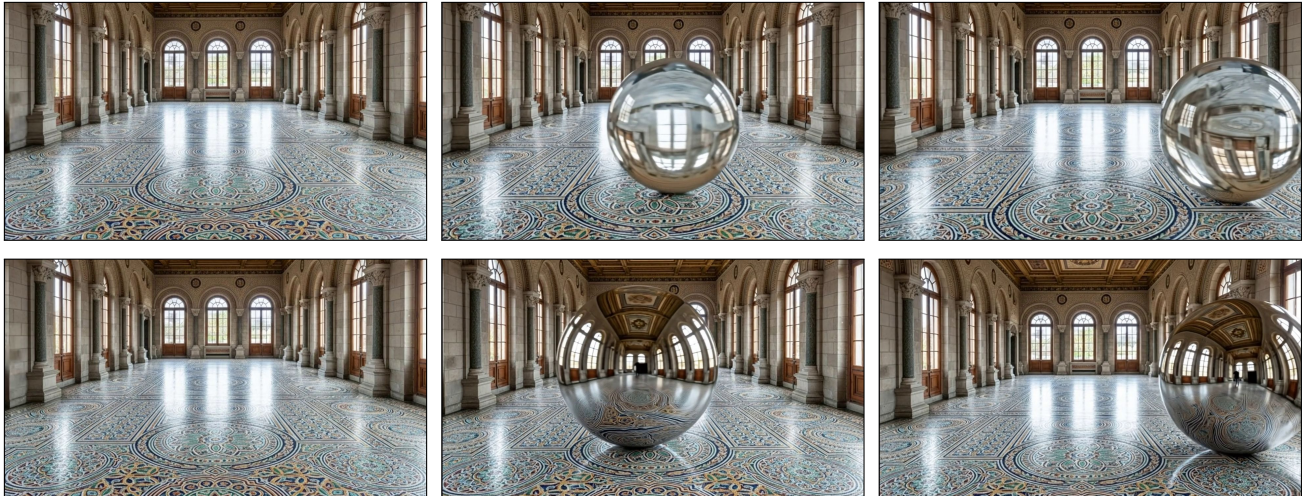


Figure 24. **Material optics. Glass** (top). Prompt: “A giant glass sphere rolls through the room. Static camera, no pan, no zoom, no dolly.” Note that the image through the glass sphere is inverted. Success rate: 0.96. **Mirror** (bottom). Prompt: “A giant mirror-polish metal sphere rolls through the room. Static camera, no pan, no zoom, no dolly.” Note that the image reflected off the sphere is not inverted). Success rate: 1.0.

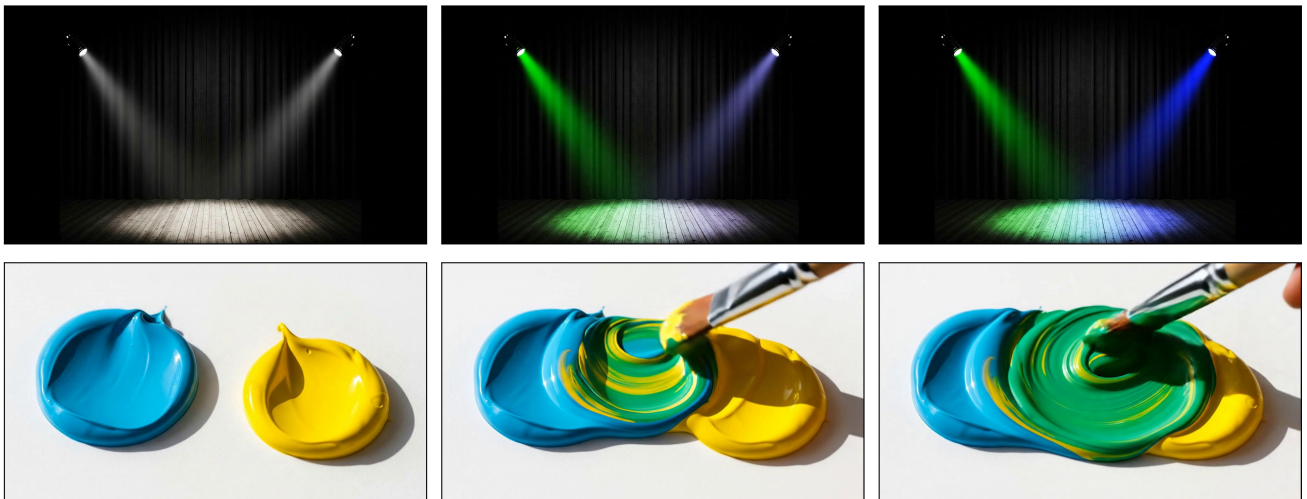


Figure 25. **Color mixing. Additive** (lights, top). Prompt: “The spotlight on the left changes color to green, and the spotlight on the right changes color to blue.” Success rate: 0.96. **Subtractive** (paints, bottom). Prompt: “A paintbrush mixes these colors together thoroughly until they blend completely. Static camera, no pan, no zoom.” Success rate: 0.86.



Figure 26. **Categorizing objects.** Prompt: “A person puts all the kids toys in the bucket. Static camera, no pan, no zoom, no dolly.” Success rate: 0.23.

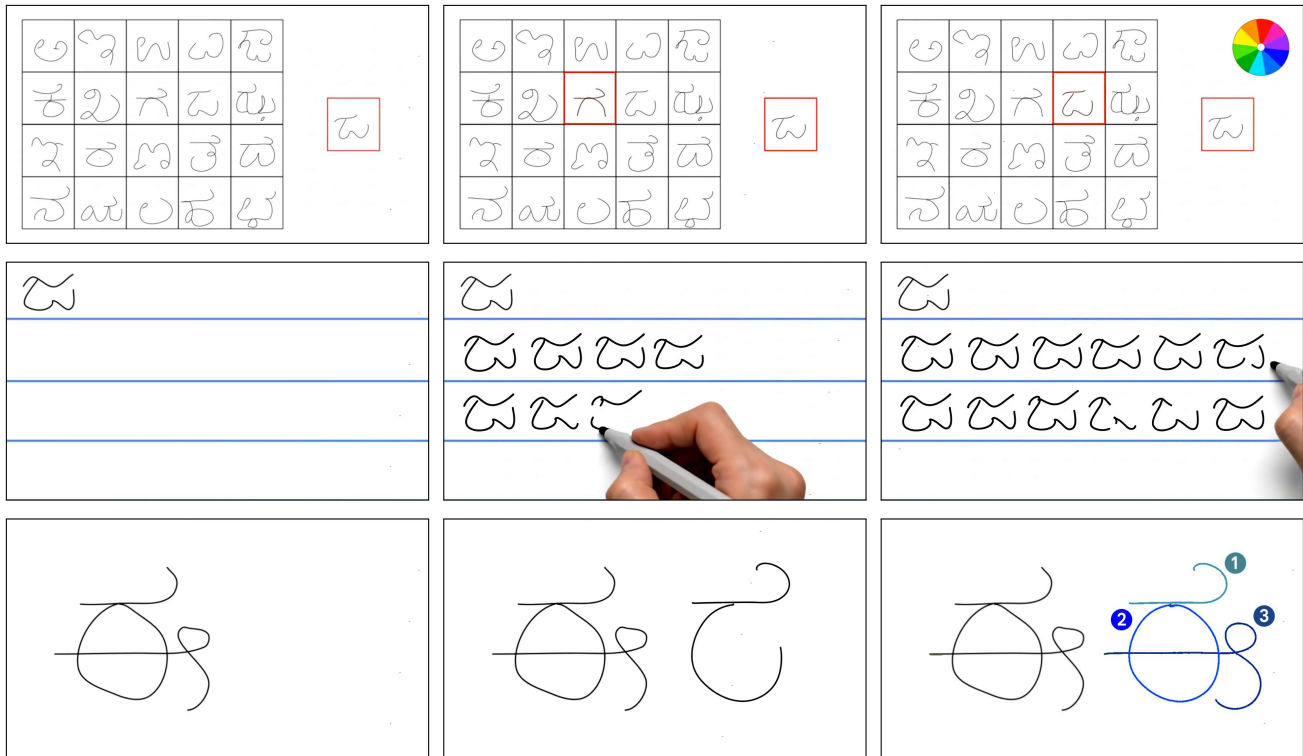


Figure 27. **Character recognition, generation, and parsing**, inspired by the Omniglot dataset (Lake et al., 2015). **Recognition** (top). Prompt: “The background of the grid cell with the same symbol as the one indicated on the right turns red. All other grid cells remain unchanged. After that, a spinning color wheel appears in the top right corner.” (Note: Veo 3 has a prior to keep things moving, which is detrimental for tasks where the solution is obtained in an early frame. We observe that a ‘motion outlet’, such as a color wheel, can indicate task completion and ‘freeze’ the solution.) Success rate: 0.41. **Generation of variations** (middle). Prompt: “The page is filled line-by-line with hand-written practice variations of the symbol.” Success rate: 0.24. **Parsing into parts** (bottom). Color and numbers in final frame are added post-hoc to show stroke order. Prompt: “Stroke-by-stroke, a replica of the symbol is drawn on the right.” Success rate: 0.36.



Figure 28. **Memory of world states**. Prompt: “The camera zooms in to give a close up of the person looking out the window, then zooms back out to return to the original view.” Success rate: 0.92.

B.3. Manipulation



Figure 29. **Background removal.** Prompt: “The background changes to white. Static camera perspective, no zoom or pan.” Success rate: 0.89.



Figure 30. **Style transfer.** Prompt: “The scene transforms into the style of a Hundertwasser painting, without changing perspective or orientation; the macaw does not move. Static camera perspective, no zoom or pan.” Success rate: 0.82.

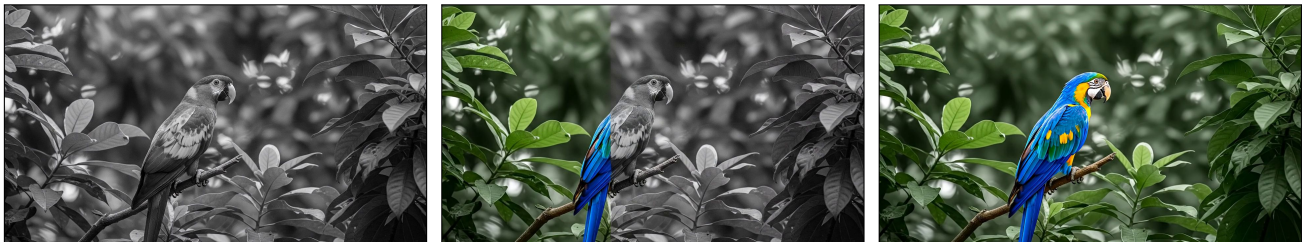


Figure 31. **Colorization.** Prompt: “Perform colorization on this image. Static camera perspective, no zoom or pan.” Success rate: 0.32.

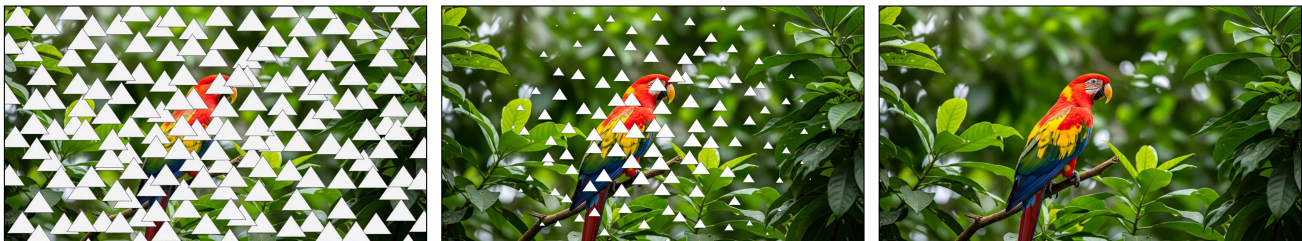


Figure 32. **Inpainting.** Prompt: “The white triangles become smaller and smaller, then disappear altogether. Static camera perspective, no zoom or pan.” Success rate: 1.0.

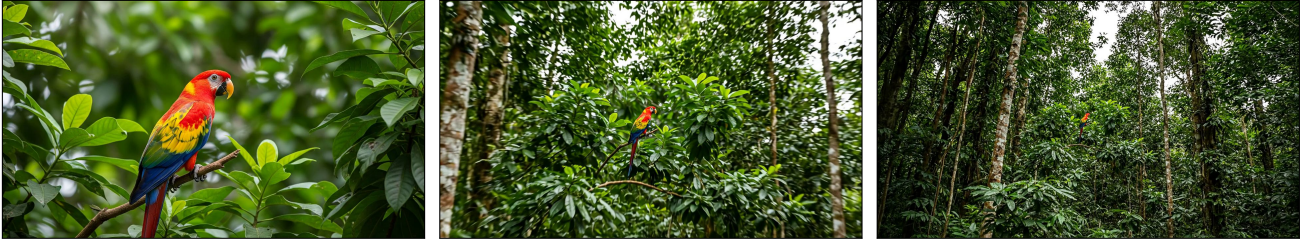


Figure 33. **Outpainting.** Prompt: “Rapidly zoom out of this static image, revealing what’s around it. The camera just zooms back, while the scene itself and everything in it does not move or change at all, it’s a static image.” Success rate: 1.0.



Figure 34. **Text manipulation.** Prompt: “Animation of the text rapidly changing so that it is made out of different types of candy (top left text) and pretzel sticks (bottom right text). Static camera perspective, no zoom or pan.” Success rate: 0.60.



Figure 35. **Image editing with doodles.** Prompt: “Changes happen instantly.” Success rate: 0.95.

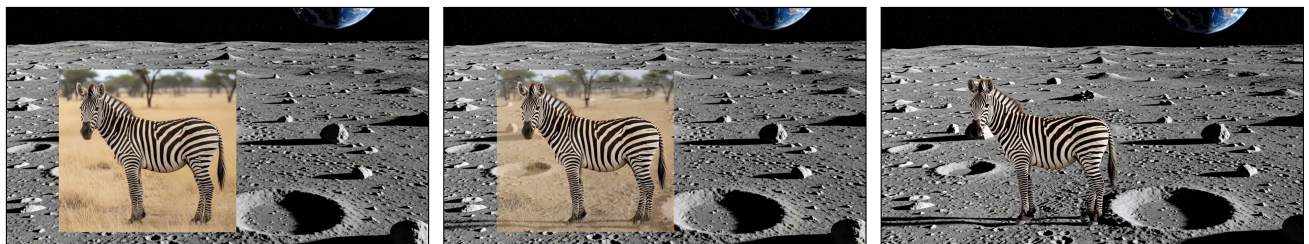


Figure 36. **Scene composition.** Prompt: “A smooth animation blends the zebra naturally into the scene, removing the background of the zebra image, so that the angle, lighting, and shading look realistic. The final scene perfectly incorporates the zebra into the scene.” Success rate: 0.81.

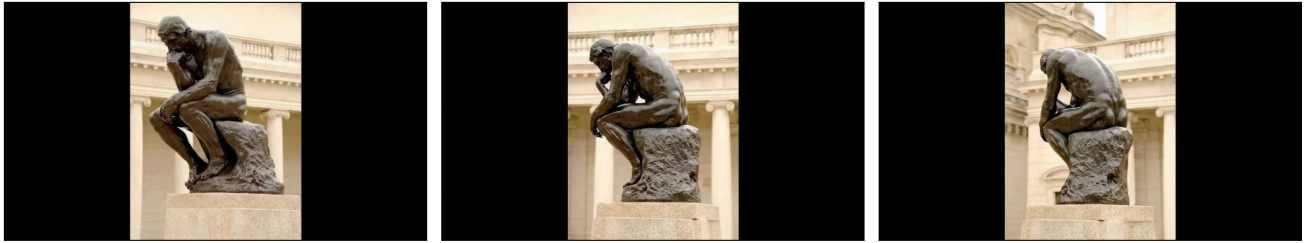


Figure 37. **Single-image novel view synthesis.** Prompt: “Create a smooth, realistic animation where the camera seems to rotate around the object showing the object from all the sides. Do not change anything else. No zoom. No pan.” Success rate: 0.96. Image source: (Deitke et al., 2023).



Figure 38. **3D-aware reposing.** Prompt: “The knight turns to face to the right and drops on one knee, lifting the shield above his head to protect himself and resting the hilt of his weapon on the ground.” Success rate: 0.54.



Figure 39. **Transfiguration.** Prompt: “A magical spell smoothly transforms the structure of the teacup into a mouse.” Success rate: 0.30.

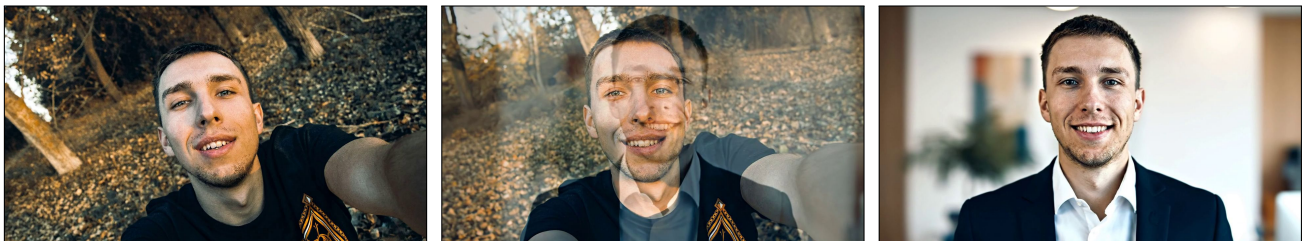


Figure 40. **Professional headshot generation.** Prompt: “Turn this selfie into a professional headshot for LinkedIn.” Success rate: 0.43. Image credit: photo by George Pisarevsky on Unsplash.

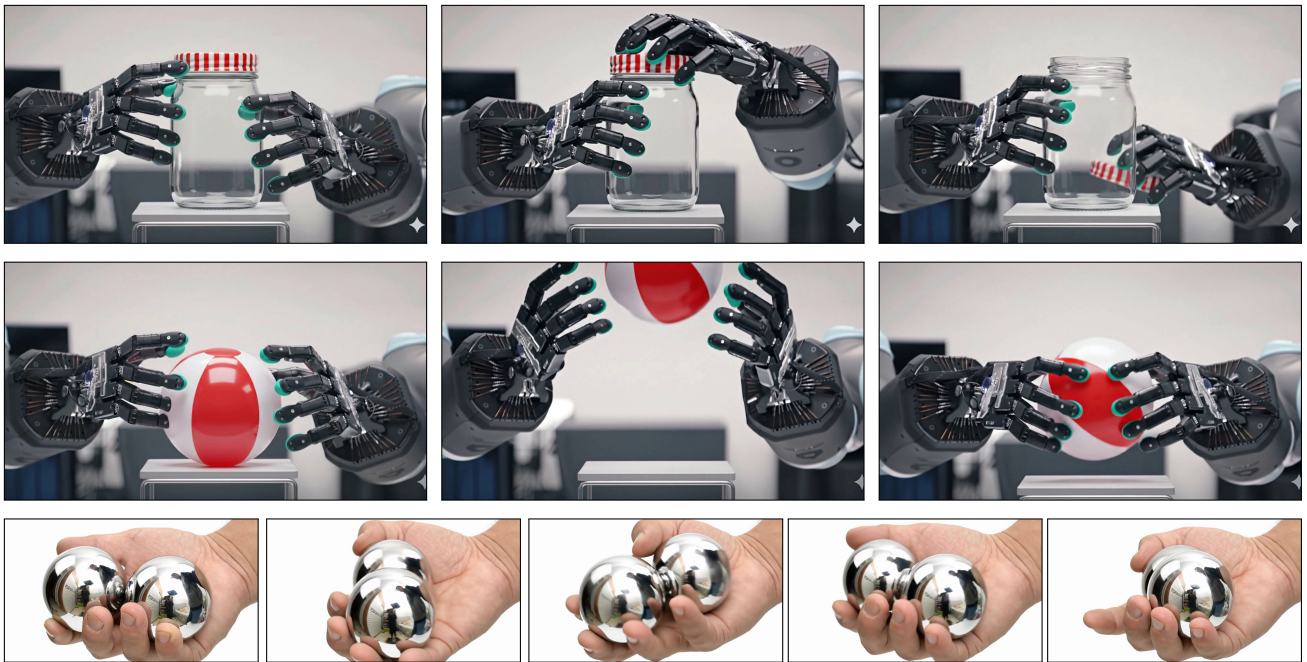


Figure 41. **Dexterous manipulation. Jar opening** (top). Prompt: “Use common sense and have the two robot hands attached to robot arms open the jar, like how a human would.” Success rate: 0.95. **Throwing and catching** (middle). Prompt: “Use common sense and have the two robot hands attached to robot arms throw the ball in the air; the ball goes up off the screen, hands move to positions to catch the ball, and catch the falling ball, like how a human would.” Success rate: 0.74. **Rotating Baoding balls** (bottom). Prompt: “A human hand holds two metal Baoding balls. The fingers, including the thumb, index, and middle finger, skillfully manipulate the balls, causing them to rotate smoothly like two planets orbiting around each other and continuously in the palm, one ball circling the other in a fluid motion.” Success rate: 0.09.

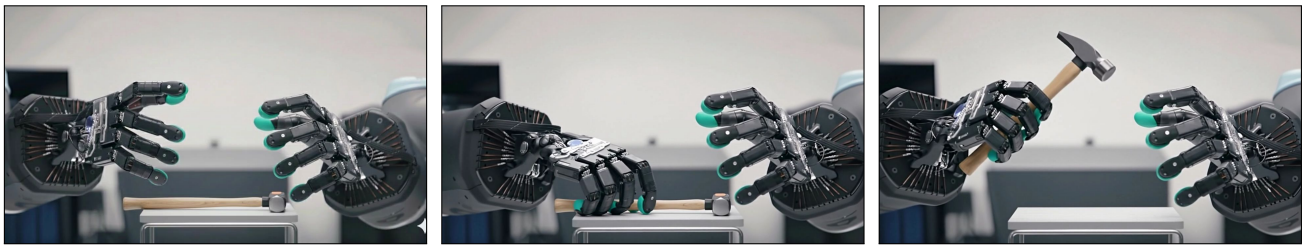


Figure 42. **Affordance recognition**. Prompt: “The robot hands mounted on robot arms pick up the hammer, naturally like how a human would.” Success rate: 0.68.



Figure 43. **Drawing**. Prompt: “A person draws a square. Static camera, no pan, no zoom, no dolly.” Success rate: 0.18.



Figure 44. **Visual instruction generation.** Prompt: “A montage clearly showing each step to roll a burrito.” Success rate: 0.42. Inspiration: (Yang et al., 2024) and Reddit.

B.4. Reasoning

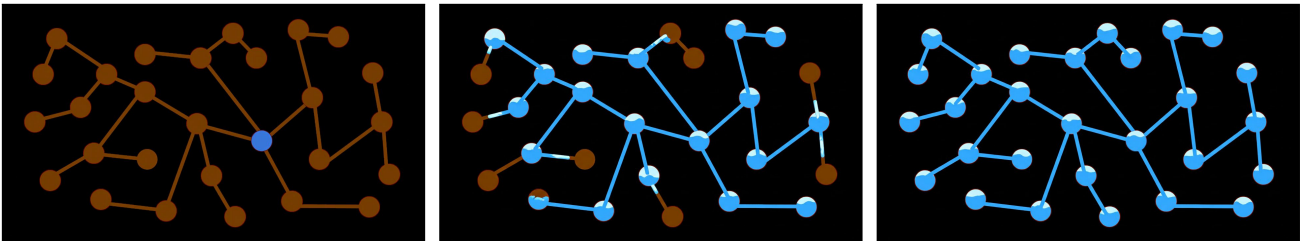


Figure 45. **Graph traversal.** Prompt: “Starting from the blue well, an unlimited supply of blue water moves through the connected channel system without spilling into the black area.” Success rate: 0.09.

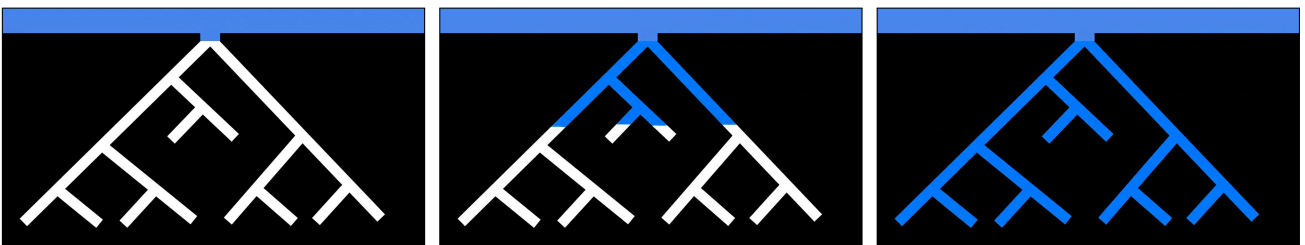


Figure 46. **Tree BFS.** Prompt: “From the blue water basin, an unlimited supply of water flows at constant speed into the cave system until all caves are filled. Static camera perspective, no zoom or pan.” Success rate: 0.27.

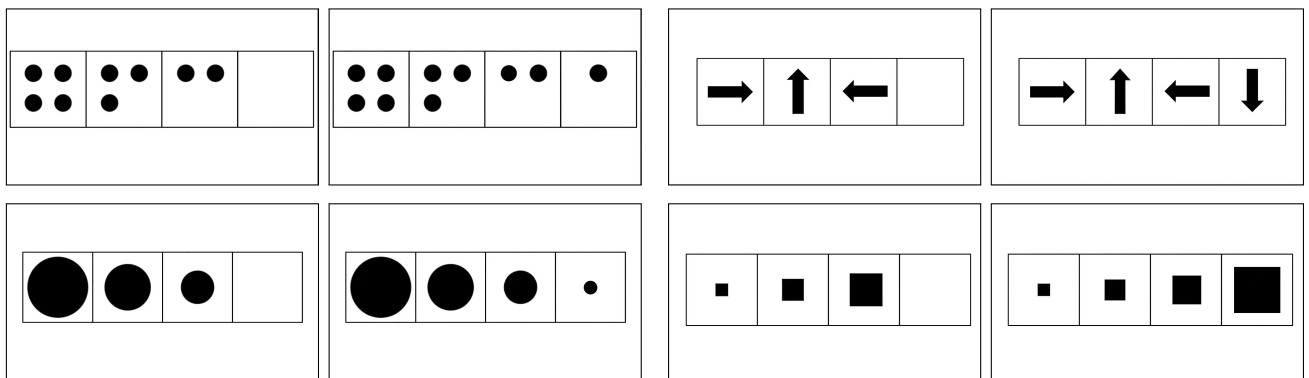


Figure 47. **Sequence completion** inspired by Raven’s progressive matrices. Each of the four pairs shows input (left) and generated output (right). Prompt: “Draw the figure that completes the pattern in the rightmost box. The images in the boxes are static. Do not modify the existing images, only draw in the empty box. Static camera, no zoom, no pan, no dolly.” Success rate: 0.21 for dots, 1.0 for arrows, 0.86 for shrinking circles, 0.74 for growing squares.

Video models are zero-shot learners and reasoners

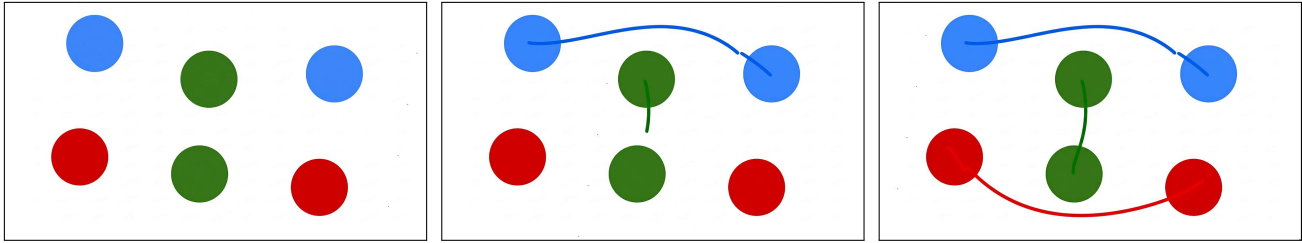


Figure 48. **Connecting colors.** Prompt: “Draw three curves, one connecting each pair of circles of the same color.” Success rate: 0.14.

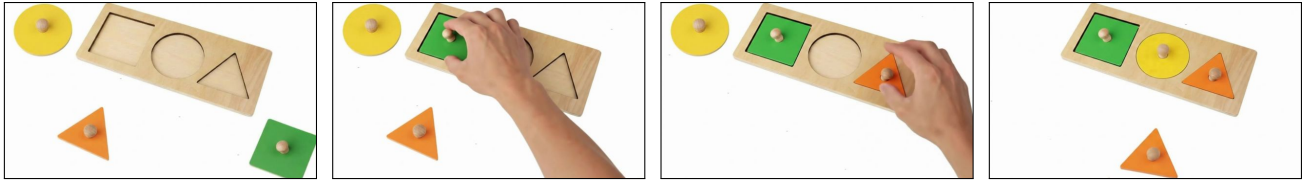


Figure 49. **Shape fitting.** Prompt: “The scene shows three colored pieces, and a wooden panel with three holes. Each colored piece fits into one and only one hole. A hand grabs each colored piece and puts it into an empty hole that has the exact same shape - if it doesn’t fit, the hand tries another hole. All the objects must be placed in their respective holes.” Success rate: 0.18.

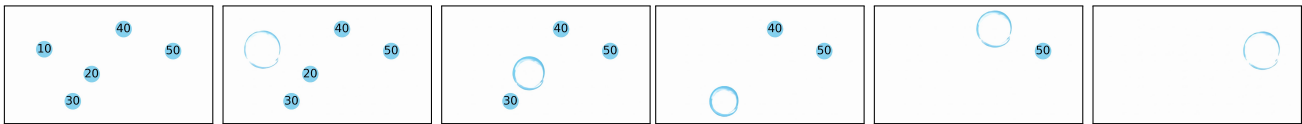


Figure 50. **Sorting numbers.** Prompt: “The video starts with some numbered bubbles. The bubbles pop and disappear one at a time, in numeric order, starting from the one with the smallest number.” Success rate: 0.13.



Figure 51. **Tool use.** Prompt: “A person retrieves the walnut from the aquarium.” Success rate: 0.92 (retrieval via tool) and 0.04 (retrieval via tool without intersecting the glass).

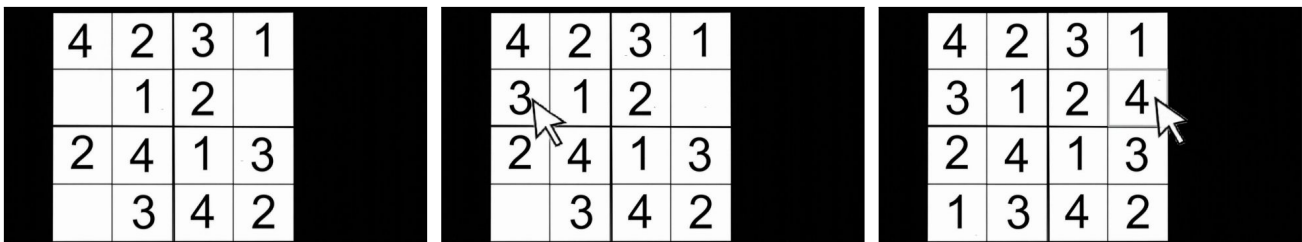


Figure 52. **Simple Sudoku completion.** Prompt: “Create a static, smooth, animation that solves the given 4x4 sudoku. Enter the missing numbers one by one. Do not change anything else in the picture. Only fill the numbers in the empty cells so the sudoku is solved properly. A cursor moves and fills the correct number in the empty boxes.” Success rate: 0.73.

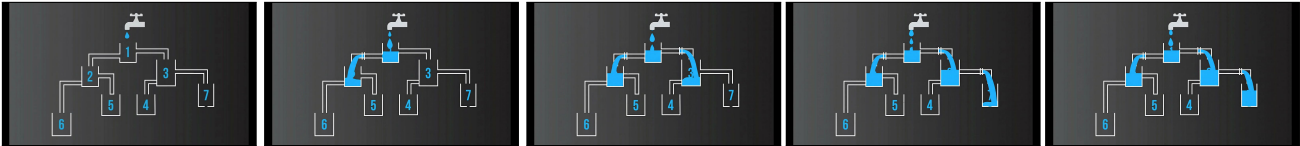


Figure 53. **Water puzzle solving.** Prompt: “The tap is turned on and water starts flowing rapidly filling the containers. Create a smooth, static animation showing the containers getting filled with water in the correct order.” (Note: Not all containers can be filled since some pipes are closed off, such as the pipe connecting container 2 to container 5. Veo fills the correct containers in the right order.) Success rate: 0.27.



Figure 54. **Maze solving.** Prompt: “Without crossing any black boundary, the grey mouse from the corner skillfully navigates the maze by walking around until it finds the yellow cheese.” Success rate: 0.09.



Figure 55. **Robot navigation.** Prompt: “The robot drives to the blue area. Static camera perspective, no movement no zoom no scan no pan.” Success rate: 0.56. Image credit: [Micromelon Robotics website](#) with permission from Tim Hadwen.

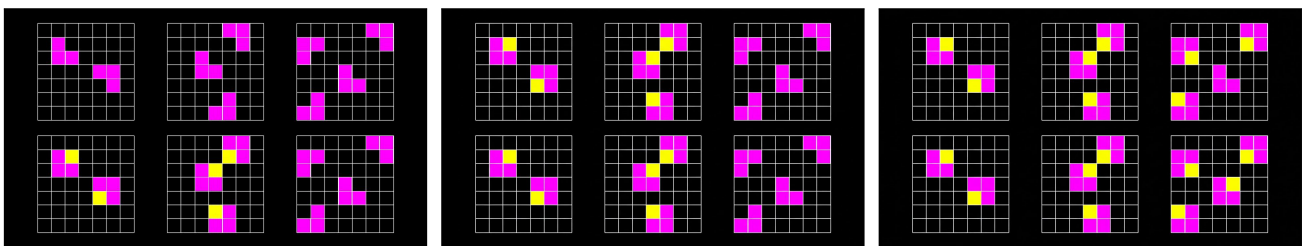


Figure 56. **Rule extrapolation** inspired by ARC-AGI (Chollet, 2019). Prompt: “Modify the lower-right grid to adhere to the rule established by the other grids. You can fill cells, clear cells, or change a cell’s color. Only modify the lower-right grid, don’t modify any of the other grids. Static scene, no zoom, no pan, no dolly.” Success rate: 0.09. While Veo 3 doesn’t follow the prompt perfectly, the output grid (bottom right) is completed correctly.

B.5. Evaluation details

For each task, we generate 22 completions with Veo 3 based on the same prompt and initial frame. Twelve completions are annotated by one of the authors, another ten samples are each annotated by two authors, and one external computer vision expert. All annotations follow the general and specific instructions outlined below. The final success rate is computed as the average success over all samples, where the success rate for a sample with multiple annotations is the average of its annotations.

We track the pair-wise inter-rater reliability between annotators for the ten samples using Cohen’s kappa (Cohen, 1960). κ is 0.82 between authors and 0.87 and 0.89 between each author and the external annotator, compared to an analytical upper bound of 1.0 (Geirhos et al., 2020b). This is considered *strong to almost perfect* agreement (Cohen, 1960; McHugh, 2012), indicating a stable and objective evaluation.

Annotators were given the following general instructions:

- Indicate whether the generation fulfills the task outlined in the prompt (1) or not (0).
- For tasks where the target is a single image (e.g., edge detection, deblurring, ...), indicate 1 if there is at least one frame that fulfills the task.
- For tasks where the target is a video segment (e.g., robot navigation, ...), indicate 1 if there is a video segment that fulfills the task.
- After the task has been completed, subsequent hallucinations or other artifacts are okay and don’t invalidate the answer.
- While prompts may specify a static camera perspective, small camera movements are okay (e.g., zoom, pan, ...).
- Hallucinations or glitches are okay if they do not interfere with the task objective.
- Very small animations or local changes are okay if the task is otherwise fulfilled correctly.
- You can ignore any audio.

Additionally, annotators had the following specific instructions for some tasks

- **Background removal.** Some branches can be considered background or foreground.
- **Buoyancy (bottle cap).** Does the bottle cap float up after being submerged?
- **Buoyancy (rock).** Does the stone sink?
- **Categorizing objects.** Are all and only kids toys put in the bucket?
- **Color mixing (additive).** Is the overlapping light cyan?
- **Color mixing (subtractive).** Is the mixed paint green?
- **Dalmation illusion understanding.** Does the video reveal the Dalmatian?
- **Deblurring.** It is okay for the background to remain slightly out of focus.
- **Dexterous manipulation (jar opening).** The entire interaction should be plausible.
- **Dexterous manipulation (throwing and catching).** The entire interaction should be plausible.
- **Image editing with doodles.** Ignoring the speed at which the changes happen, are the changes that are introduced the requested ones (hat, snow, scarf)?
- **Drawing.** Is there a complete square, and each line is drawn with the pen? Lines can be drawn or re-drawn in any order.
- **Edge detection.** Does the model produce a plausible edge-map of the scene?
- **Graph traversal.** Are the edges/nodes filled in in the correct order? Not all edges/nodes have to be filled in during the 8s video. Score as failure if water appears somewhere without flowing from the initial well.
- **Gravity and air resistance (on earth).** Does the feather fall slower than the ball due to air resistance?
- **Gravity and air resistance (on the moon).** Do the feather and the ball fall at the same rate?

- **Professional headshot generation.** Is the likeness of the person preserved?
- **Material properties.** Does the paper burn in a plausible manner?
- **Memory of world states.** Does the scene in the end match the scene in the beginning?
- **Single-image novel view synthesis.** It's fine if the rotation isn't 360 degrees.
- **Character recognition.** At the moment the spinning color wheel appears, is only the correct symbol highlighted?
- **Character generation of variations.** Are the new characters plausible, hand-written variations of the original?
- **Character parsing into parts.** Is the character decomposed into plausible strokes?
- **Material optics (glass).** Is the image in the sphere inverted?
- **Material optics (mirror).** Is the image in the sphere right-side up?
- **3D-aware reposing.** Is there a frame showing the knight in the correct final pose, while maintaining his appearance? Implausible transformations are okay as long as the final pose is correct.
- **Physics body transform (rigid body).** Does the vase retain its shape and appearance?
- **Physics body transform (soft body).** Does the scarf drape plausibly over the vase?
- **Robot navigation.** Does the robot drive to the blue area in a physically plausible manner? (If e.g. a wall is knocked over by the robot, that could be plausible as well, as long as it could happen in reality)
- **Rule extrapolation.** Is the lower-right grid completed correctly? Modifications to the other grids are okay.
- **Scene composition.** The zebra does not have to maintain the pose from the image, but should be included plausibly in the scene (e.g., not hovering above the ground).
- **Segmentation.** Does the model produce a plausible segmentation of some key objects in the scene, while not introducing new objects?
- **Sequence completion.** Only focusing on the rightmost box and ignoring potential changes elsewhere, is the correct shape drawn?
- **Shape cue-conflict understanding.** Does the video show a family of cats?
- **Shape fitting.** Is there a moment when all holes are filled with the correct shape?
- **Style transfer.** Does the style of the image change to a painting?
- **Simple sudoku completion.** Are the correct numbers filled in? Top-left: 3, top-right: 4, bottom-left: 1. It is okay if other numbers are modified.
- **Tool use.** Is the walnut retrieved from the aquarium in a plausible manner?
- **Tree BFS.** Are the branches filled in a plausible manner (top to bottom)? Not all branches have to be filled in during the 8s video.
- **Visual instruction generation.** Are all necessary steps shown? A time-lapse/montage is okay.
- **Visual jenga.** Are the objects removed in a plausible order? Weird hand movements or object deformations are okay as long the order is plausible and objects are not reintroduced.
- **Water puzzle solving.** Are only the correct containers filled in, and in the right order 1-2-3-7?

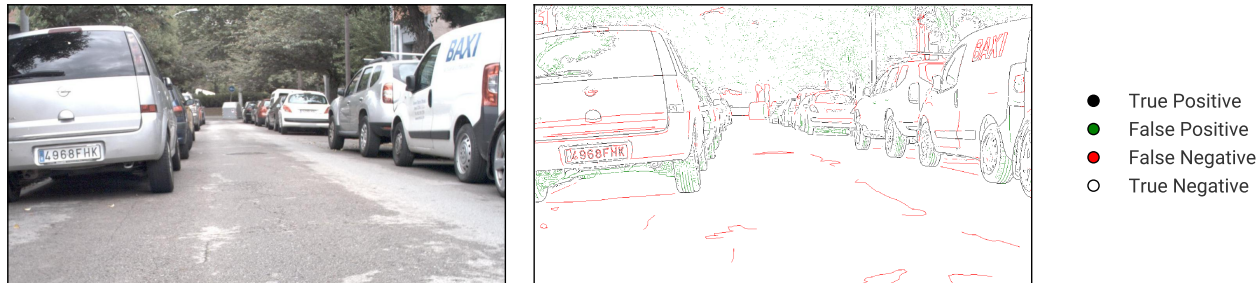


Figure 57. **Graded Veo 3 edge map.** While false negatives reflect genuine oversights of Veo 3 (e.g., cracks in the road, lettering on the car), many false positives correspond to actual image details that seem to be erroneously excluded from the ground truth (e.g., the outline of the trees, the reflection in the car window, and the tire profiles).

C. Quantitative results: experimental details

Table 1. **Video count breakdown for quantitative tasks.** For segmentation, 2×1 splits indicate one test set with two different background color prompts (white/green). For the prompt sensitivity study on the symmetry task (App. D), 2×10 splits indicate 2 splits (random/shape) across 10 tested prompt variations. In total, there are 884 different starting images for the quantitative tasks, and 46 for the qualitative tasks (if all macaw-based images are counted as a single image). For the qualitative tasks, we additionally generated 1364 videos ($62 \text{ tasks} \times 22 \text{ samples}$).

Task	Splits	Imgs/split	Pass@	Video models	Total videos
Edge	1	50	10	2	1000
Segmentation	2×1	50	10	2	2000
Object extraction	1	54	10	2	1080
Editing	1	30	1	2	60
Maze	2×4	50	10	2	8000
Symmetry	2	25	10	2	1000
Symmetry prompt analysis	2×10	25	1	1	500
Analogy	4	50	10	2	4000
Total					17640

C.1. Perception: Edge detection

We provide details for the image editing task in Sec. 3.1.

Evaluation As is standard in the literature, we refine and binarize predicted edges and allow for small local shifts compared to the ground truth (Dollár & Zitnick, 2013; 2014; Zitnick & Dollár, 2014; Leng et al., 2024). Concretely, we use non-maximum suppression, then binarize with one of 16 evenly-spaced thresholds, then thin the binary edge map. At each threshold, we find the optimal mapping between predicted and ground-truth edge pixels within a radius of 0.75% of the image diagonal (around 11 pixels). Fig. 57 shows an example rating of a Veo 3-generated edge map. We report the best OIS over k attempts (optimal image scale; the maximum F_1 -score over all thresholds) for the best/last frame.

Dataset We used all 50 test images from BIPEDv2 (Soria et al., 2020; 2023).

Models & prompts We tested Veo 3 `veo-3.0-generate-preview` and Veo 2 `veo-2.0-generate-preview-001` through the Vertex AI API. We also tested Nano Banana `gemini-2.5-flash-image-preview` through Google AI Studio.

Veo

All edges in this image become more salient by transforming into black outlines. Then, all objects fade away, with just the edges remaining on a white background. Static camera perspective, no zoom or pan.

Nano Banana

Outline all edges in the image in black, make everything else white.

Sampling We generated 10 videos per sample with a fixed prompt.

C.2. Manipulation: Object extraction

We provide details for the image editing task in Sec. 3.2.

Evaluation We extract the last frame from each generated video; the resulting image is converted to greyscale, a binary mask with threshold 200 is applied, and the number of connected components is extracted using `scipy.ndimage.label`, resulting in the count estimate. We also report the chance baseline which can be calculated as: $\text{random} - \text{chance} = 1 - (1 - p)^k$ where p is the probability to get the count correct via guessing (here: $p = \frac{1}{9}$) and $k \in [1, 10]$.

Dataset We generated an animal counting dataset using Nano Banana. Starting from a white 16:9 image, we used the following prompt, where `number` is in $[1, 9]$ and `animal` is in `['dog', 'elephant', 'cat', 'brown bear', 'horse', 'rabbit', 'raccoon']`. We manually evaluated the generated dataset for correctness; the resulting dataset has 54 images (exactly 6 per count).

Nano Banana

Exchange the white space with a realistic photograph of: exactly {number} {animal}, outside, not overlapping, in a natural landscape.

Models & prompts We tested Veo 3 `veo-3.0-generate-preview` and Veo 2 `veo-2.0-generate-preview-001` through the Vertex AI API.

Veo

The background changes to white. Then:

- *If there is just a single animal: the animal sits in the middle of the image, looking straight at the camera.*
- *If there are multiple animals: all animals line up in a row, with ample white space between them.*

Sampling We generated 10 videos per sample with a fixed prompt.

C.3. Reasoning: Maze solving

We provide details for the image editing task in Sec. 3.3.

Evaluation Our evaluation process is tailored to the model type. For Veo, we analyze the generated video frame-by-frame, extracting the path taken by the agent (red circle). We check for any invalid moves, such as jumping over walls, clipping through boundaries, or any alteration of the goal’s position. We report the success rate as the fraction of k attempts where the agent successfully reaches the goal (green circle) without any illegal moves.

For Nano Banana, which generates the full path in one edit, we assess whether the drawn path connects the start and end points (allowing for minor discontinuities) and crucially, whether it intersects with any maze walls or goes off the valid path.

For Gemini 2.5 Pro with a maze input as an image (I2T) or as ASCII (T2T), we check whether the series of grid positions represents an uninterrupted path from the start position to the goal.

Dataset For rectangular mazes, we generated 50 random mazes per size using `maze-dataset` (Ivanitskiy, 2025), but replacing the square start and end with circles and swapping their colors. We also drew 10 irregular mazes by hand and flipped/rotated them to obtain 40 unique samples.

Models & prompts We tested Veo 3 `veo-3.0-generate-preview` and Veo 2 `veo-2.0-generate-preview-001` through the Vertex AI API. We also tested Nano Banana `gemini-2.5-flash-image-preview` and Gemini 2.5 Pro `gemini-2.5-pro` through Google AI Studio.

Veo

Create a 2D animation based on the provided image of a maze. The red circle slides smoothly along the white path, stopping perfectly on the green circle. The red circle never slides or crosses into the black areas of the maze. The camera is a static, top-down view showing the entire maze.

Maze:

- *The maze paths are white, the walls are black.*
- *The red circle moves to the goal position, represented by a green circle.*
- *The red circle slides smoothly along the white path.*
- *The red circle never slides or crosses into the black areas of the maze.*
- *The red circle stops perfectly on the green circle.*

Scene:

- *No change in scene composition.*
- *No change in the layout of the maze.*
- *The red circle travels along the white path without speeding up or slowing down.*

Camera:

- *Static camera.*
- *No zoom.*
- *No pan.*
- *No glitches, noise, or artifacts.*

Gemini 2.5 Pro I2T

SYSTEM

Think step by step as needed and output in xml format:

<think>thinking process</think>

<final_answer>final answer</final_answer>

USER

The following image shows a maze, represented by colored squares:

- *Black squares represent walls and cannot be passed through.*
- *White squares are empty and can be passed through.*
- *The red square is the starting point.*
- *The green square is the end point.*

Please solve the maze by providing a path from the starting point to the end point. The path should be provided as a list of coordinates of each step, where each coordinate is a (row, col) tuple, and row, col are 0-based indices. Consider the origin (0, 0) to be the top-left corner. Overall, the path should be provided in the format of [(row1, col1), (row2, col2), ...].

A valid path must:

- *Start at the starting point (the red square).*
- *End at the end point (the green square).*
- *Avoid the walls (the black squares).*
- *Pass only through empty space (the white squares).*
- *Move one square at a time.*
- *Only move up, down, left, and right, not diagonally.*

Correct your answer if you spot any errors.

Here is the maze image: {image}

Nano Banana

Mark the correct path from the red to the green circle through the maze in blue.

Gemini 2.5 Pro T2T

SYSTEM

*Think step by step as needed and output in xml format:**<think>thinking process</think>**<final_answer>final answer</final_answer>*

USER

The following is an ASCII-representation of a maze:

- ‘#’ represents walls which cannot be passed through.
- ‘ ’ represents empty spaces that can be passed through.
- ‘S’ is the starting point.
- ‘E’ is the end point.

Please solve the maze by providing a path from the starting point to the end point. The path should be provided as a list of coordinates of each step, where each coordinate is a (row, col) tuple, and row, col are 0-based indices. Consider the origin (0, 0) to be the top-left corner. Overall, the path should be provided in the format of [(row1, col1), (row2, col2), ...].

A valid path must:

- Start at the starting point ‘S’.
- End at the end point ‘E’.
- Avoid the walls ‘#’.
- Pass only through empty space ‘ ’.
- Move one square at a time.
- Only move up, down, left, and right, not diagonally.

*Correct your answer if you spot any errors.**Here is the maze in ASCII format: {maze}*

Sampling We generated 10 videos per sample with a fixed prompt. Note that for Gemini 2.5 Pro I2T, we represented the maze as a grid where the red and green positions are marked as squares (not circles) to make the setup grid-like (i.e., a matrix with cells), since this might be easier for a language model.

C.4. Reasoning: Visual symmetry solving

We provide details for the visual symmetry task in Sec. 3.4.

Evaluation We prompt Veo with input images containing a 10×16 grid where a pattern is drawn on the left half. The goal is to complete the pattern on the empty right half so that the final pattern is symmetrical along the central vertical axis.

We compare Veo’s best-frame and last-frame solutions with the ground-truth symmetrical grid and compute the number of incorrectly-colored cells. A cell is determined as incorrectly-colored if the average color across pixels in the cell is perceptually distinct from the ground-truth average color in the matching cell. We compute perceptual color differences of the average cell color in the CIELAB color space, with a difference threshold of 15.0. In Fig. 5, we report the percentage of attempts in which the best or last frame solution has zero incorrect cells for $k = 1$.

Dataset We created a synthetic grid coloring image dataset to evaluate visual symmetry. We generated 25 samples using common symmetrical symbols, objects and shapes such as english letters (e.g., A, H, M, X), geometric shapes (e.g., square, triangle), symmetrical objects (e.g., wineglass, balloon; together, the *shape* condition). We also generated 25 samples consisting of randomly-colored cells (the *random* condition).

Models & prompts We tested Veo 3 `veo-3.0-generate-preview` and Veo 2 `veo-2.0-generate-preview-001` through the Vertex AI API. We also tested Nano Banana `gemini-2.5-flash-image-preview` through Google AI Studio.

Veo

Instantly reflect this pattern along the central, vertical axis while keeping the existing colored pattern without modification. Static camera perspective, no zoom or pan.

Sampling We generated 10 videos per sample with a fixed prompt.

D. Prompting best practices

Table 2. **Prompt sensitivity study on the visual symmetry task.** We report best frame pass@1 % and the average number of incorrectly-colored cells across 25 samples on each split (shape/random).

No.	Prompt	Pass@1		Avg incorrect cells	
		Shape	Random	Shape	Random
1	Instantly reflect this pattern along the central, vertical axis while keeping the existing colored pattern without modification.	48	68	4.16	7.00
2	Instantly reflect this pattern along the central, vertical axis while keeping the existing colored pattern without modification. Static camera perspective, no zoom or pan.	42	65	5.00	3.52
3	Instantly reflect this pattern along the central, vertical axis while keeping the existing colored pattern without modification. The result needs to be mirror-symmetrical along the vertical axis. Static camera perspective, no zoom or pan.	36	52	6.28	9.04
4	One by one, cells in the right half of the grid are filled in to complete the pattern. The pattern is mirror-symmetrical along the central vertical line. Static shot; no zoom, no pan, no dolly.	32	12	10.76	14.08
5	Reflect this pattern along the central, vertical axis.	28	40	9.76	4.52
6	An animation showing the left half of the grid being mirrored onto the right half to create a symmetrical pattern. Static shot; no zoom, no pan, no dolly.	24	12	10.96	16.32
7	You’re a master symmetry solver. Your task is to fill the cells on the right side of the grid to mirror the pattern on the left, such that it’s symmetrical along the vertical axis.	24	8	9.20	17.72
8	Fill color in the appropriate cells on the right side of the grid to complete the pattern. The final image should be symmetrical along the central vertical line. Static shot, no zoom no pan no dolly.	13	9	10.30	14.74
9	Create a static, smooth, realistic animation completing the pattern in the image by filling the grid on the right hand side. Do not change anything else. No zoom, no pan.	12	4	14.88	21.00
10	A timelapse of a professional pixel artist drawing a symmetrical pattern onto a white canvas. Static shot; no zoom, no pan, no dolly.	8	20	14.20	12.64

Results are best-effort estimates of Veo’s performance using carefully chosen prompts. Generally, performance varies greatly with the exact task description provided in the prompt, as illustrated by a prompt sensitivity study on the visual symmetry task in Table 2. Here are best practices from this sensitivity analysis and our other experiments:

- **Remove ambiguity.** Tasks can be solved in a variety of ways, and natural language descriptions tend to leave a lot of room for interpretation. The goal should be formulated clearly, e.g., saying “symmetrical along the central, vertical axis”, rather than just “symmetrical”.
- **Specify what shouldn’t change.** Veo has a tendency to change any part of the input to create interesting, dynamic scenes. Including not only a positive task description, but also specifying what *not* to change can help mitigate this, e.g., “keep the existing colored pattern without modification”.
- **Providing an outlet.** As mentioned above, Veo has a strong prior to keep things moving. Providing a “motion outlet” in the form of, e.g., a spinning ball can help keep the rest of the scene static.
- **Let the model decide when its done.** The motion prior also means that Veo often keeps modifying the scene, even after solving the task. Providing a visual indicator, e.g., “add a glowing red dot once the goal is reached” allows for easy extraction of the solution from the generated video.
- **Scene and camera controls.** Phrases like “static camera, no zoom, no pan, no dolly” can help keeping the scene static, e.g., for image-to-image tasks.
- **Speed control.** Some tasks like maze solving benefit from being solved step-by-step. For other tasks, especially image-to-image tasks, specifying instant changes can help avoid artifacts.

- **Realism.** Veo was trained to generate plausible, realistic-looking videos. Translating an abstract task into a realistic setting (including, but not limited to editing the original image to depict realistic, 3D scenes rather than abstract shapes) can improve generation results. A similar effect was observed in (Jeong et al., 2025), and we expect *visual prompt engineering* to emerge as a powerful tool for video models.

E. Veo inference details

For both qualitative and quantitative tasks, we generate videos via the Vertex AI API. We do not manually specify a seed, making generation non-deterministic and subject to Vertex’s default sampling behavior. All videos are generated without audio and default to a length of eight seconds.

At the time of writing, generation costs 0.20 USD/s for Veo 3 and 0.50 USD/s for Veo 2.

Video generation took 18.2s on average per video; this includes not only the time to generate the video but also the potential waiting time until resources become available and the computation is scheduled, as well as the time it takes to transfer the input from our local machine to the datacenter and the output video back to the local machine. In principle, all of these steps can be parallelized (e.g., when running a larger dataset), subject to network bandwidth and compute availability.

F. Failure cases



Figure 58. **Monocular depth estimation.** Prompt: “The image transitions to a depth-map of the scene: Darker colors represent pixels further from the camera, lighter colors represent pixels closer to the camera. The exact color map to use is provided on the right side of the image. Static scene, no pan, no zoom, no dolly.” Failure: Veo 3 seems generally unable to color pixels by depth beyond a binary foreground/background mapping and specifically struggles with using a provided color map.

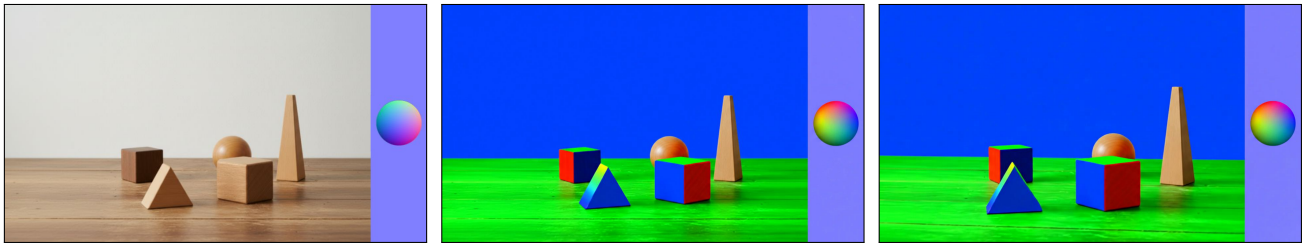


Figure 59. **Monocular surface normal estimation.** Prompt: “The image transitions to a surface-normal map of the scene: the red/green/blue color channel specify the direction of the surface-normal at each point, as illustrated on the right side of the image on a sphere. Static scene, no pan, no zoom, no dolly.” Failure: While Veo 3 shows some promise in coloring surfaces according to their orientation (e.g., the cube in the front), coloration is inconsistent (compare the two cubes) and doesn’t correctly interpolate colors (e.g., for the slope on the triangle).



Figure 60. **Force & motion prompting**, inspired by (Gillman et al., 2025; Geng et al., 2024). **Force prompting** (top). Prompt: “The balls move in the direction indicated by the arrows. Balls without an arrow don’t move. Static scene, no pan, no zoom, no dolly.” **Motion trajectory prompting** (bottom). Prompt: “Each car drives out of the frame following the indicated trajectory. Static camera, no zoom, no pan, no dolly.” Failure: Veo 3 seems unable to follow force/motion annotations with any consistency. Providing annotations for the first frame and letting the model remove them before generating the scene in motion does not work, either.

Video models are zero-shot learners and reasoners

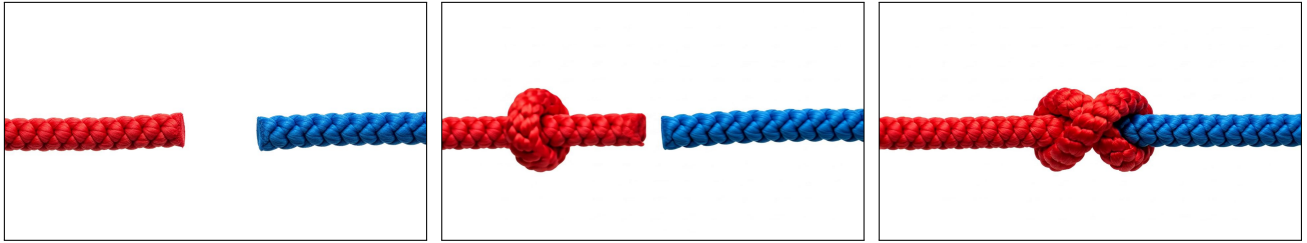


Figure 61. **Tying the knot.** Prompt: “A knot is tied connecting these two rope ends.” Failure: physics violation, impossible rope movement.

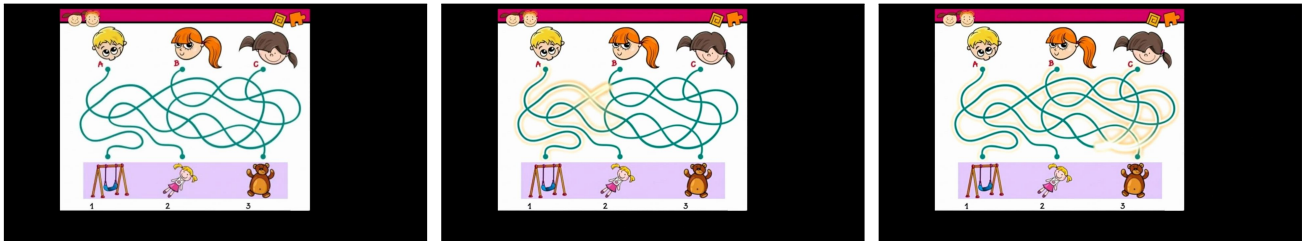


Figure 62. **Connect the path puzzle.** Prompt: “The path connecting the boy to the object starts glowing slowly. Nothing else changes. No zoom, no pan, no dolly.” Failure: hallucinations, lighting up of all paths.

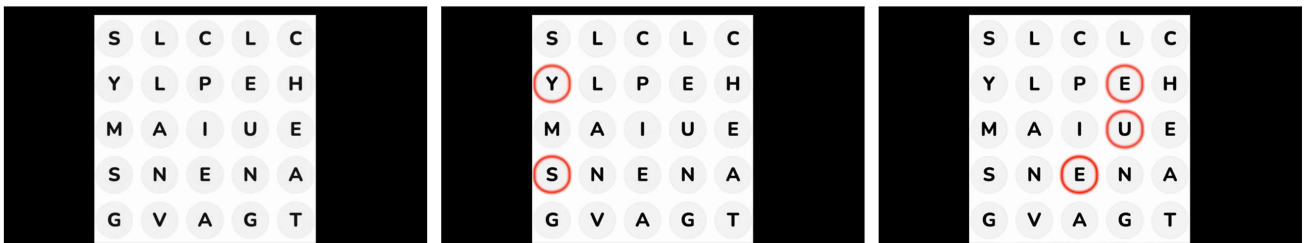


Figure 63. **Five letter word search.** Prompt: “Generate a static video animation using the provided letter grid. The task is to highlight the only 5-letter English word CHEAT, which may be oriented in any direction (horizontally, vertically, or diagonally). The animation should consist of a semi-transparent red rectangle with rounded corners smoothly fading into view, perfectly encapsulating the five letters of the word. The rectangle should have a subtle, soft glow. Do not change anything else in the image. The camera must remain locked in place with no movement. No zoom, no pan, no dolly.” Failure: does not recognize words; highlights individual letters randomly.

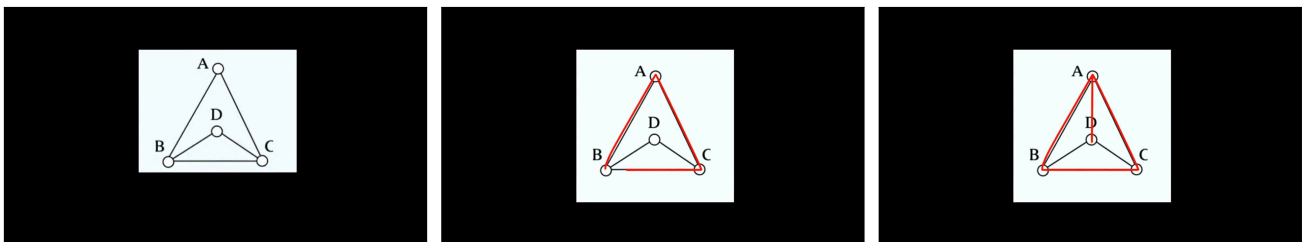


Figure 64. **Eulerian path.** Prompt: “Create a smooth animation where a red pen traces all existing edges in a continuous path without lifting the pen. All edges need to be traced. Do not visit any edge twice and do not lift the pen. No zoom, no pan.” Failure: does not trace the edges exactly, traces non-existent edges.

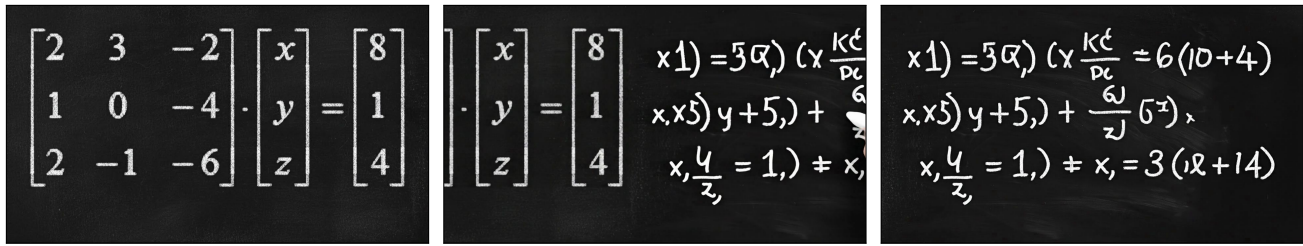


Figure 65. **Solving system of linear equations.** Prompt: “A hand appears and solves the set of linear equations. It replaces the x, y, z matrix with their correct values that solves the equation. Do not change anything else.” Failure: hallucinations with text on the blackboard.



Figure 66. **Spot the difference.** Prompt: “There are two images. The left image is different from the right image in 5 spots. Create a static, realistic, smooth animation where a cursor appears and points at each place where the left image is different from the right image. The cursor points one by one and only on the left image. Do not change anything in the right image. No pan. No zoom. No movement. Keep the image static.” Failure: does not identify all the differences. Hallucinates differences.

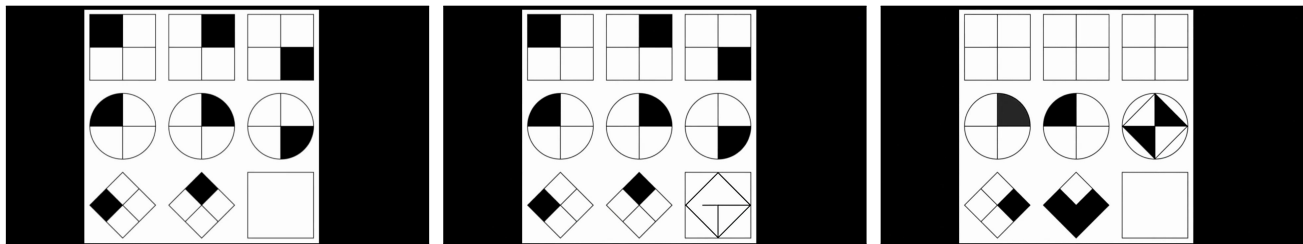


Figure 67. **Visual IQ test.** Prompt: “Create a static, smooth, animation that solves the puzzle in the given image. The correct pattern should appear at the bottom right to solve the puzzle. Do not change anything else in the picture. No zoom, no pan, no dolly” Failure: incorrect figure pattern.



Figure 68. **Glass falling.** Prompt: “The object falls. Static camera, no pan, no zoom, no dolly.” Failure: physics violation, glass does not break, and orients itself to be vertical after landing on the floor.

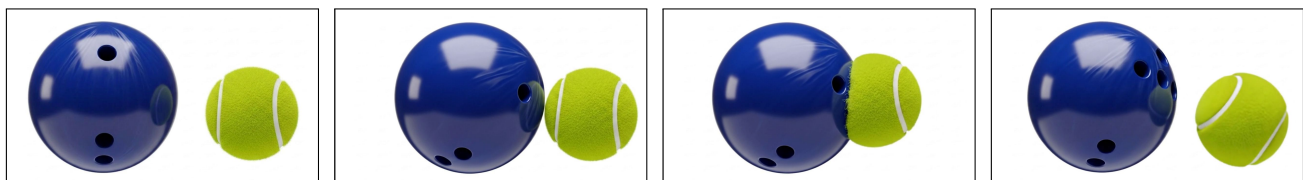


Figure 69. **Collisions.** Prompt: “The two objects collide in slow motion. Static camera, no pan, no zoom, no dolly.” Failure: not physically plausible, the objects pause at the moment of impact and then are pushed together by an invisible force.

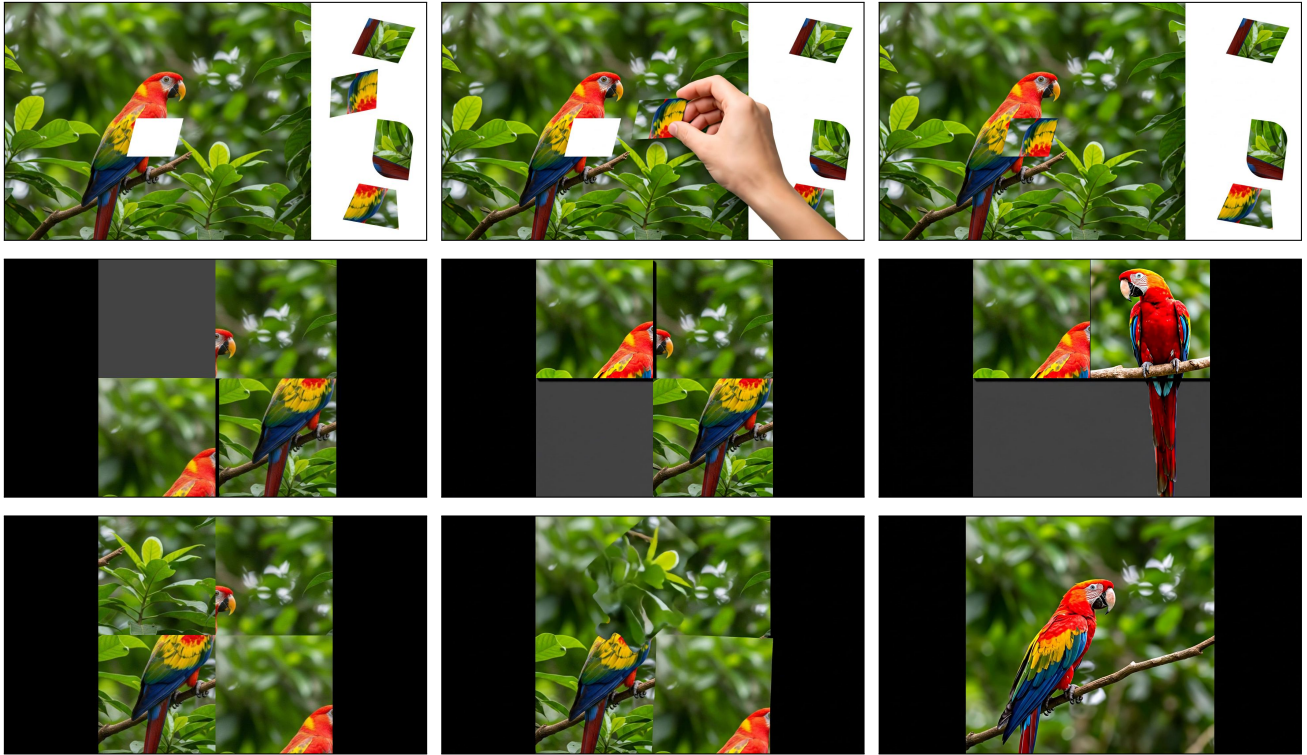


Figure 70. **Tiling puzzles. Jigsaw puzzle** (top). Prompt: “A hand takes the fitting puzzle piece from the right, rotates it to be in the correct orientation, then puts it into the hole, completing the puzzle. Static scene, no pan, no zoom, no dolly.” Failure: wrong piece orientation. **Sliding puzzle** (middle). Prompt: “Slide the pieces of this sliding puzzle around one-at-a-time until all edges align.” Failure: doesn’t maintain piece integrity while sliding, hallucinates new pieces. **Scrambled puzzle** (bottom). Prompt: “Unscramble this image.” Failure: image details are inconsistent with original pieces.

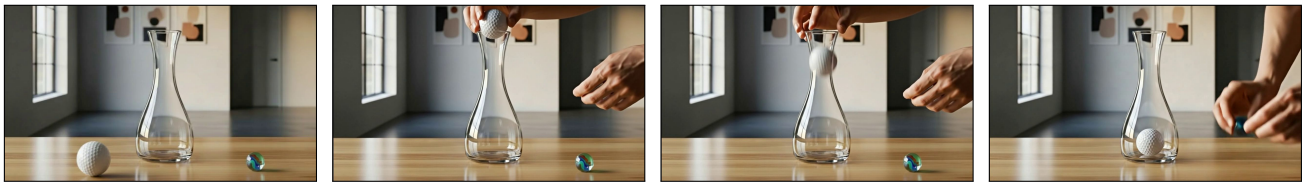


Figure 71. **Bottleneck**. Prompt: “A person tries to put the golf ball in the vase. Static camera, no pan, no zoom, no dolly.” Failure: not physically plausible, golf ball is too large to pass through the bottleneck of the vase.



Figure 72. **Laundry folding**. Prompt: “Generate a video of two metal robotic arms properly folding the t-shirt on the table. Failure: physics violation, implausible folding movements.

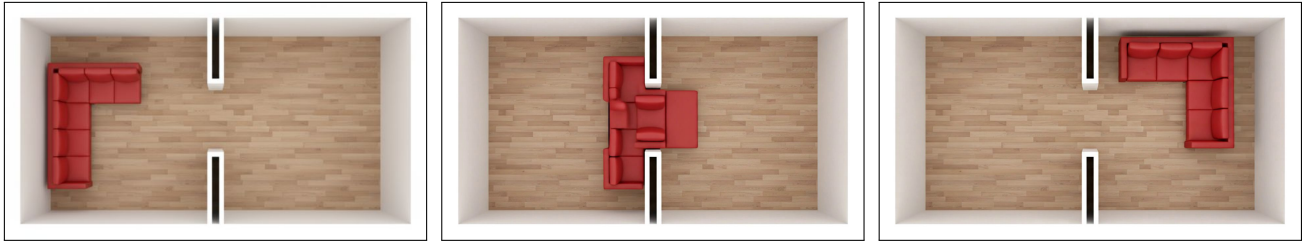


Figure 73. **Motion planning**; inspired by the piano mover’s problem. Prompt: “The red couch slides from the left room over into the right room, skillfully maneuvering to fit through the doorways without bumping into the walls. The walls are fixed: they don’t shift or disappear, and no new walls are introduced. Static camera, no pan, no zoom, no dolly.” Failure: violating rigid-body integrity, not keeping to permissible transformations (rotation, translation).

G. Similarities and differences with NLP

There are strong reasons to believe that progress in vision will mirror the rapid advancements seen in NLP. Both language and vision models are developed on the same hardware infrastructure, and thus benefit from the same hardware scaling laws. Additionally, many pivotal algorithmic breakthroughs, such as the Transformer architecture, are modality-agnostic and have been successfully adapted for both domains. We therefore expect inference costs for vision models to decrease just as they are decreasing for LLMs.

However, there are also crucial differences. A significant hurdle for vision is the cost of training on information-dense video data, which could slow the adoption of large-scale vision models compared to their NLP counterparts. Conversely, data availability presents a long-term advantage for vision. It is widely believed that we are approaching the limits of available high-quality text data, whereas the amount of video data being generated globally continues to grow exponentially. This suggests that while training costs are a near-term challenge, the vast and growing reservoir of video data may ultimately fuel the development of even more powerful foundation models for vision.

H. LLM use

Gemini 2.5 Flash and Gemini 2.5 Pro (Comanici et al., 2025) were used for brainstorming task ideas, suggesting related work that we might have otherwise missed, coding support, and to polish human writing.

I. Image sources

Where not stated in the figure caption, images were obtained as follows.

- Figs. 7 to 12, 29 to 35 and 70: The original macaw image was generated with Gemini and, depending on the figure, subsequently modified by the authors (e.g., conversion to grayscale, adding noise, adding the monkey with Nano Banana).
- Fig. 13: The input image was obtained from [here](#) (Apache 2.0 license) based on the LOLv2 dataset (Yang et al., 2021) and randomly selected. The image was slightly cropped to fit a 16:9 aspect ratio.
- Figs. 14, 18 to 21, 23 to 26, 28, 36, 38, 39, 43, 44, 49, 51, 61, 65, 68, 69 and 71 to 73: generated with Gemini.
- Fig. 22: The input image was obtained from [here](#) (CC0 license).
- Fig. 27: hand drawn by us, inspired by Fig. 1 of the Omniglot paper (Lake et al., 2015).
- Fig. 37: sample from Objaverse (Deitke et al., 2023)
- Figs. 45 to 48, 50, 52 and 54: created by us.
- Figs. 53, 62, 63 and 66: original image from Reddit.
- Fig. 56: hand drawn by us, inspired by ARC-AGI (Chollet, 2019).

- Fig. 57: sample from BIPEDv2 ([Soria et al., 2020; 2023](#)).
- Figs. 58 to 60: generated with Gemini, then annotated by us.
- Figs. 64 and 67: hand drawn by us. Inspired by original images from Reddit.
- Figs. 41 and 42: The robot hands are extracted from a frame in [this video](#) and were subsequently adapted with Nano Banana. The hands holding Baoding balls were obtained from [here](#).