iWISDM: Assessing instruction following in multimodal models at scale

Xiaoxuan Lei * [1] [2] Lucas Gomez * [3] [2] Hao Yuan Bai * [4] [2] Pouya Bashivan [1] [2]

Abstract

The ability to perform complex tasks from detailed instructions is key to the remarkable achievements of our species. As humans, we are not only capable of performing a wide variety of tasks but also very complex ones that may entail hundreds or thousands of steps to complete. Large language models and their more recent multimodal counterparts that integrate textual and visual inputs have achieved unprecedented success in performing complex tasks. Yet, most existing benchmarks are largely confined to single-modality inputs (either text or vision), narrowing the scope of multimodal integration assessments, particularly for instruction-following in multimodal contexts. To bridge this gap, we introduce the instructed-Virtual VISual Decision Making (iWISDM) environment, engineered to generate a limitless array of vision-language tasks of varying complexity. Using iWISDM, we compiled three distinct benchmarks of instruction-following visual tasks across varying complexity levels and evaluated several newly developed multimodal models on these benchmarks. Our findings establish iWISDM as a robust benchmark for assessing the instructional adherence of both existing and emergent multimodal models and highlight a large gap in these models' ability to follow instructions.

1. Introduction

A typical day in most people's lives involves hundreds or thousands of tasks. Most of which are performed without explicit attention. Just in between getting up and getting to work, one may have already performed 5-15 tasks (taking a shower, shaving, making coffee, getting dressed, etc.). Teaching artificial agents to perform similar seemingly mundane tasks has proven to be an extremely difficult computational problem (Konar, 2018). The challenge becomes more apparent when one realizes that each of these seemingly mundane tasks such as making coffee involves performing tens of steps/actions (Appendix A Figure. 3). The challenge becomes increasingly more significant once we consider more complex tasks such as operating a device or assembling a piece of furniture from its instruction manual. And yet, these tasks are performed proficiently by most individuals in most situations.

Large Multimodal Models (LMMs) offer a possible avenue for solving multimodal multi-step decision-making tasks. Several benchmarks have been developed to evaluate these models. However, the existing benchmarks for assessing such models face several shortcomings: **(1)** Most multimodal benchmarks like FLEURS (audio-based, (Conneau et al., 2023)) and VATEX (video-based, (Wang et al., 2019)) are still unimodal in their inputs, and do not permit detailed assessment of models' capacity to integrate information across modalities towards task goals. **(2)** Visual Question-Answering (VQA) datasets like VQAv2 (Goyal et al., 2017) and CLEVR (Johnson et al., 2017) assess reasoning with visual information in static images without addressing temporal information integration and sequential decision-making. **(3)** Open-ended learning environments such as XLand (Team et al., 2021), Crafter (Hafner, 2021), and Minecraft (Guss et al., 2019) have been utilized for training reinforcement learning agents. It remains unclear whether and how they can be adapted to benchmark LMMs. **(4)** To the best of our knowledge, none of the existing benchmarks specifically assess models' ability to precisely follow instructions in decision-making tasks, an important measure of reliability and trustworthiness. Despite its importance, conducting such assessments has been particularly challenging because of the difficulty of collecting samples of multi-step tasks with ground truth information. **(5)** More recent benchmarks such as MME (Fu et al., 2023), Mmbench (Liu et al., 2023c) and MMvet (Yu et al., 2023) cover a wide range of cognitive tasks that start to adopt manually-generated or GPT-powered responses. However, those benchmarks are difficult to scale, which makes them

[*] Equal contribution [1] Department of Physiology, McGill University, Montréal, Canada [2] Mila, University of Montréal, Montréal, Canada [3] Integrated Program in Neuroscience, McGill University, Montréal, Canada [4] School of Computer Science, McGill University, Montréal, Canada. Correspondence to: Pouya Bashivan <pouya.bashivan@mcgill.ca>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

inconvenient when investigating the scaling properties of LMMs. In addition, benchmarks such as OwlEval (Ye et al., 2023) and LVLMeHub (Xu et al., 2023) rely on subjective human responses that often show high variability across individuals.

To address this gap in evaluation, we designed the **i**nstructed-**V**irtual **Vis**ual **D**ecision **M**aking (iWISDM), a virtual environment that enables procedural generation of complex, multi-step decision-making tasks that test an agent's capacity to process visual information guided by natural language instructions.

Our main contributions are:

- We introduce iWISDM, a virtual environment for the procedural generation of limitless visual decision-making tasks accompanied by natural language instructions.
- We use iWISDM to construct three vision-language multimodal benchmarks with varying complexity levels to probe LMMs' ability to follow natural language instructions.
- We test recently developed LMMs and human subjects on these benchmarks and identified a notable shared weakness across existing LMMs compared to humans in their ability to precisely follow instructions in the context of visual decision-making tasks.

2. Methodology

We developed iWISDM to streamline the procedural generation of diverse visual reasoning tasks, which vary in complexity and require minimal user intervention. These tasks engage executive functions, such as action inhibition, working memory, attentional set, task switching, and schema generalization (Fuster, 2015; Sun et al., 2023), traditionally associated with the prefrontal cortex, a critical area for advanced cognitive processes in the brain. The iWISDM task space is also designed to accommodate working memory and decision-making tasks commonly studied in neuroscience and cognitive science research (Rigotti et al., 2013; Goldman-Rakic, 1992; Fuster, 2009).

Inspired by prior work (Yang et al., 2018), iWISDM generates task trials in 3 steps: **(1)** task graph construction **(2)** node initialization **(3)** trial instantiation. Task graphs provide the general framework for defining an iWISDM task, node initialization further specifies task details, and the last step, trial instantiation, generates the trial instance based on the initialized task graphs. See Figure 1 for examples of operators and task graphs. For detailed task construction procedures, see Appendix A.2.

At its core, iWISDM is designed to be scalable and extensible. Its tasks can be readily decomposed into simpler subtasks that involve fewer sensory observations and cognitive operations. We focus on 2 types of compositionality: **logical compositionality** where decisions can be combined hierarchically through logical boolean operators, and **temporal compositionality** where tasks require multiple decisions to be made in sequence or in parallel.

Another important feature of iWISDM is **the vastness of its task space**. This feature is enabled via the logical and temporal compositionality during task generation. The ability to produce a large number of distinct tasks is important for the robust evaluation of large multimodal models that are trained on increasing volumes of information from the web.

Natural language provides a rich and convenient way of communicating complex information to biological or artificial agents. It has been shown that improvements in language understanding in LLMs directly enhance their generality (performing many tasks) and adaptation (0-shot generalization) (Brown et al., 2020; Radford et al., 2019). Therefore, each task in iWISDM is accompanied by a simplified **natural language instruction**.

In contrast to prior virtual environments tailored for cognitive or neuroscience investigations (Molano-Mazon et al., 2022), iWISDM can not only generate hand-crafted cognitive tasks such as classical decision-making tasks (e.g. contextual decision-making and n-back, Appendix Figure 1), but also allow procedural task generation from a user pre-specified task space.

We envision the future of iWISDM as a framework further developed and expanded by the larger community of machine learning scientists, cognitive scientists, and neuroscientists. For this reason, we have designed iWISDM to be **highly customizable and extendable**. First, users can define new task operators by inheriting the `Operator` class. Second, iWISDM allows any natural image set to be used as the stimulus set. Finally, users can define arbitrary object/stimuli properties to be used by the environment during task generation.

For more details regarding features of iWISDM, please refer to Appendix A.3. The code of iWISDM is available on GitHub at `https://github.com/BashivanLab/iWISDM`.

3. Evaluation

3.1. Models & Humans

Using the `AutoTask` framework in iWISDM, we created **three benchmarks** corresponding to low, medium, and high complexity tasks. These levels are set by task generation settings such as the number of `Switch` operators, trial frames, and possible actions. For further details refer to

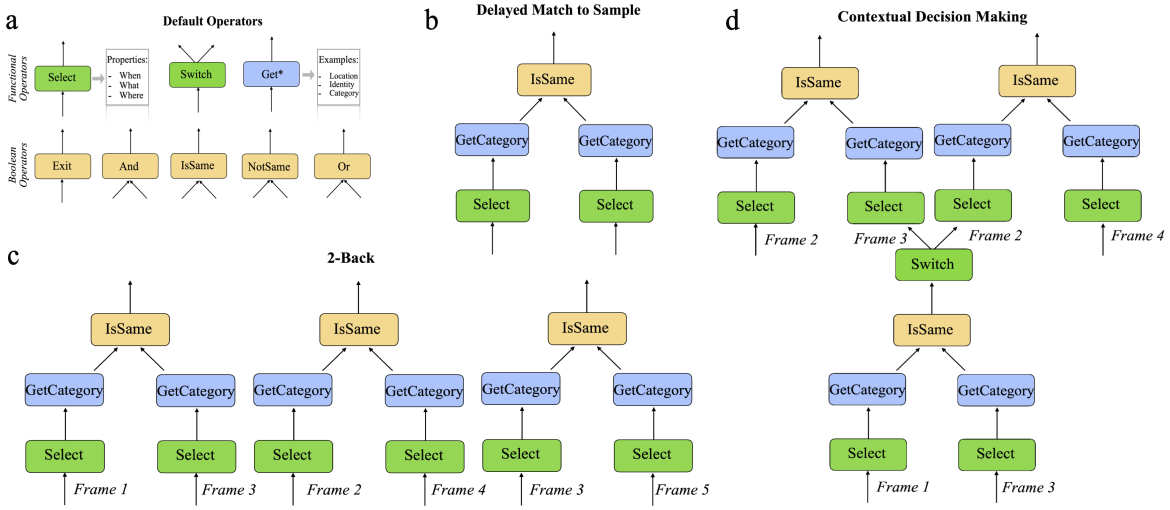


Figure 1. Description of operators and task graph examples in the core build of iWISDM. **a)** Two main types of operators are considered. Functional operators: `Select` retrieves stimuli according to specified attributes such as time and location; `Switch` accepts boolean values and directs the task logic to different subtasks; `GetAttribute` takes stimuli as input and outputs the corresponding attribute values of those stimuli. Boolean operators take boolean values as inputs and produce another boolean value based on their logic. **b-d)** Using the operators defined in panel **a**, we create tasks from pre-specified rules. Panels **b**, **c**, and **d** display the task graphs for the three typical cognitive tasks illustrated in Figure 4b, demonstrating the structures of these specific tasks in iWISDM. For instance, the task instruction for panel **d** is: *"category of object 1 equals category of object 2?"*, *"if category of object 1 equals category of object 3, then category of object 2 equals category of object 3, else category of object 2 equals category of object 4?"*. The task instruction for panel **c** is similar to panel **b** but repeated across time.

Appendix Table 1. For trial examples, refer to Appendix Figure 6, 7, 8.

As a preliminary evaluation, we test the capabilities of GPT-4V, Gemini-Pro-1.0, Claude-3 , and InternLM- XComposer2, and MMICL in solving iWISDM tasks. The set of tested models was limited due to the scarcity of applicable models. Many popular open-source LMMs, such as MiniGPT4 and LLaVa, were unable to perform the task simply due to their limited image sequence lengths. A minimum image sequence length of ten is needed to complete all complexity levels. We were able to properly evaluate two open-source models, InternLM-XComposer2-7b and MMICL-Instructblip-T5-xl. For samples of prompts used to evaluate each model see Appendix A.5.

We also collected responses from 6 human subjects tasked to answer three sets of randomly selected trials (Figure 6, 7, 8), each set sampled from a different complexity level (total of 150 trials). The task trials were displayed in a way similar to that of the models, where images can be seen alongside the task's text instruction following a general task description. We compare LMMs' performance to human baselines and find a notable gap in the multi-image instruction-following task capabilities of existing LMMs.

In addition to the above 3 benchmarks and to further investigate the performance upper bound of the selected models, we evaluated them on two sets of simple single-frame tasks (location only and category only). See Figure 11 for results.

3.2. Results

Figure 2a shows the accuracy of actions taken by the models plotted against the complexity level for each type of prompt. GPT-4V generally achieves the best model scores, with the largest performance gap on the low and high complexity tasks. However, relative to Human performance these gaps are marginal at best. Broadly, MMICL and Gemini-Pro-1.0 were the worst-performing models. The expected inverse correlation between complexity and action accuracy was only captured clearly by the GPT-4V and Gemini-Pro-1.0 results.

In contrast to model performance, human subjects scored much more accurately, with scores ranging from 0.78 to 0.98 across complexities. This model-human gap in performance indicates a significant shortcoming of LMMs on multi-image instruction following tasks. This shortcoming is unlikely due to insufficient feature understanding, as the high single-frame category task performance (Figure 2b) conflicts with the observed weak category task performances across complexities.

We also analyzed how each model performed on subsets of tasks where single or multiple object properties were

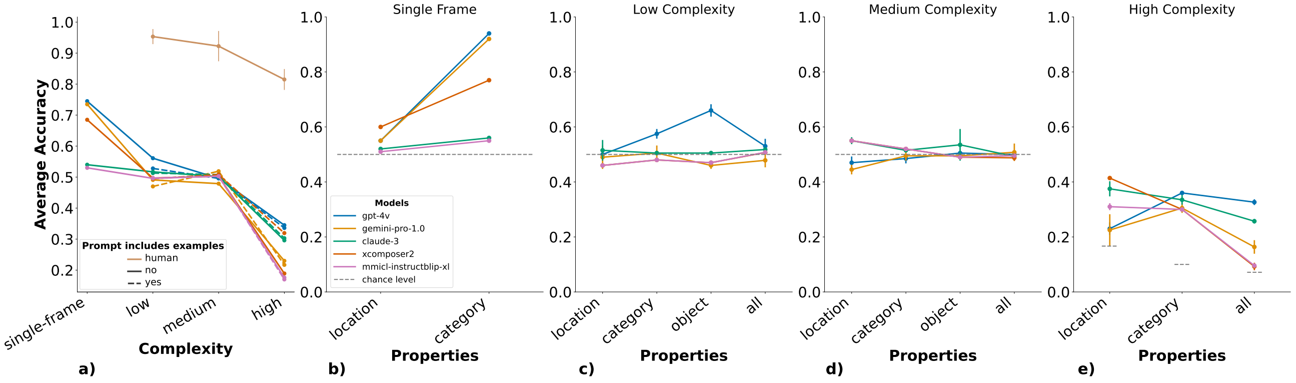


Figure 2. **a)** Average accuracy of LMMs and Humans on iWISDM benchmarks of varying complexity, and prompt type. **b), c), & d)** Average accuracy of applicable models on low, medium, and high complexity tasks for each feature type. Chance level represents the baseline accuracy of randomly guessing one of the applicable actions.

involved. Figure 2b-e shows the average accuracy of all models conditioned on task properties and complexity level. We did not include the object-identity subgroup in the high complexity plot of Figure 2e as non-boolean response types for object identity tasks are not feasible. The performances on high complexity tasks show a clear ranking between the abilities of GPT-4V and Gemini-pro models on tasks with diverse response options. For almost all models and complexities, location-only tasks posed the most difficulties. This finding confirms previous analyses of GPT-4V, which found that it often struggles to correctly recognize an object's position within an image[1] (Majumdar et al., 2024).

To further investigate the response patterns of the LMMs we performed further analyses on a subset of the models seen in Appendix Figures 11, 12, 13, and 14. To determine the effects of delay frames on model performance a set of simple delayed-match-to-sample tasks were generated which only differed in difficulty by the number of delay frames. Appendix Figure 11 shows that for InternLM-XComposer2, MMICL, and GPT-4v, the addition of single or multiple delay frames in a task has little effect on task performance. Next, we explored how the number of different boolean operators affected open-source model (InternLM-XComposer2 & MMICL) performance. The Appendix Figure 12a-d results show that, generally, across all boolean operators an increase in their abundance within a task leads to worse model performance, which is expected. In Appendix Figure 13 we examined how the number of stimuli affected task performance across complexities. We expected to see that as the number of stimuli increases, the task performance would decrease. This was found to be the case for Low complexity tasks, which can be seen in Appendix Figure 13a. However, Medium and High complexity tasks displayed an inverse trend, seen in Appendix Figure 13b and c. We were unable to find a plausible explanation for this unexpected inverse trend; as such, further analyses are required. Finally, we investigated the exact effect that different required response types had on accuracy for the High complexity benchmark. As Appendix Figure 14 shows, when tasks required a response that was a non-boolean type word, the models performed significantly worse.

4. Conclusion and Future Directions

We introduced iWISDM as a platform for validating multimodal models. As a benchmark, our primary focus is on assessing the ability of LMMs to follow instructions in visual-language decision-making tasks. We developed three benchmarks of incremental complexity and evaluated several LMMs and human performance. The gap between LMMs and humans indicates that LMMs still lack key abilities to solve instruction-following tasks. Through a detailed analysis of LMMs' behaviour patterns, we identified diminished spatial recognition ability and decreased performance with increased task complexity. We believe iWISDM will be an important benchmark that complements existing benchmarks which evaluate LMM capabilities in areas such as commonsense reasoning, numerical computation, or relational inferences (Fu et al., 2023). Future work could probe the separate components of LMMs and further identify the specific weaknesses in their instruction-following abilities. We can then address these weaknesses and improve existing models. Finally, we are in the process of fine-tuning LMM models on iWISDM and evaluating them on other benchmarks.

[1] https://blog.roboflow.com/gpt-4v-object-detection

5. Acknowledgements

X.L. was supported by CAMBAM fellowship (2024) and Doctoral Excellence Scholarship, Union Neuroscience et Artificial Intelligence Quebec, UNIQUE (2021-2022). L.G. was supported by Masters Excellence Scholarship, Union Neuroscience et Artificial Intelligence Quebec, UNIQUE (2024). This research was supported by the Healthy-Brains-Healthy-Lives startup supplement grant and the NSERC Discovery grant RGPIN-2021-03035. P.B. was supported by FRQ-S Research Scholars Junior 1 grant 310924, and the William Dawson Scholar award. All analyses were executed using resources provided by the Digital Research Alliance of Canada (Compute Canada) and funding from Canada Foundation for Innovation project number 42730. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Additionally, we would like to thank the ICML reviewers for providing valuable feedback and constructive criticism.

6. Limitations

iWISDM was evaluated on a limited number of open-source models. During the development of this work, more models were released.

The stimuli dataset used during trial instantiation was rendered from ShapeNet. It can be argued that this limits the evaluation of LMM models on naturalistic stimuli. We are in the process of extending iWISDM to employ stimuli from the COCO dataset (Lin et al., 2014). We also argue that our ShapeNet stimuli have less noise than naturalistic stimuli, so they are appropriate for evaluating model performance.

Although the current version of iWISDM cannot probe LMM functions such as numerical computation and relational inference, it could cover some of these additional capabilities by adding new operators like Count (to tally specific objects) or Relative (to identify properties like location relative to other objects).

7. Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. Apart from the many potential societal consequences of progressing in this field, we would like to acknowledge that with fast-paced progress in developing more capable LMMs, their comprehensive assessments, especially in the context of instruction following, become ever-important. iWISDM addresses this question in a scalable and customizable way. We hope iWISDM will lead to better and safer models while inspiring further analyses of LMM behaviour.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., and Zhou, J. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

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.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Chen, J., Guo, H., Yi, K., Li, B., and Elhoseiny, M. Visualgpt: Data-efficient adaptation of pretrained language models for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18030–18040, 2022.

Chen, J., Zhu, D., Haydarov, K., Li, X., and Elhoseiny, M. Video chatcaptioner: Towards the enriched spatiotemporal descriptions. *arXiv preprint arXiv:2304.04227*, 2023a.

Chen, L., Li, J., Dong, X., Zhang, P., He, C., Wang, J., Zhao, F., and Lin, D. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793*, 2023b.

Chen, Y.-C., Li, L., Yu, L., El Kholy, A., Ahmed, F., Gan, Z., Cheng, Y., and Liu, J. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pp. 104–120. Springer, 2020.

Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

Computer-Vision-in-the-Wild. Cvinw readings: A collection of papers on the topic of "computer vision in the wild (cvinw)". https://github.com/Computer-Vision-in-the-Wild/CVinW_Readings, 2024. Accessed: 2024-02-16.

Conneau, A., Ma, M., Khanuja, S., Zhang, Y., Axelrod, V., Dalmia, S., Riesa, J., Rivera, C., and Bapna, A. Fleurs: Few-shot learning evaluation of universal representations of speech. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pp. 798–805. IEEE, 2023.

Dai, W., Li, J., Li, D., Tiong, A., Zhao, J., Wang, W., Li, B., Fung, P., and Hoi, S. Instructblip: towards general-purpose vision-language models with instruction tuning. arxiv. *Preprint posted online on June*, 15:2023, 2023.

Dasgupta, I., Kaeser-Chen, C., Marino, K., Ahuja, A., Babayan, S., Hill, F., and Fergus, R. Collaborating with language models for embodied reasoning. *arXiv preprint arXiv:2302.00763*, 2023.

Driess, D., Xia, F., Sajjadi, M. S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Yang, J., Zheng, X., Li, K., Sun, X., et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

Fuster, J. *The prefrontal cortex*. Academic Press, 2015.

Fuster, J. M. Cortex and memory: emergence of a new paradigm. *Journal of cognitive neuroscience*, 21(11): 2047–2072, 2009.

Goldman-Rakic, P. S. Working memory and the mind. *Scientific American*, 267(3):110–117, 1992.

Goyal, P., Mahajan, D., Gupta, A., and Misra, I. Scaling and benchmarking self-supervised visual representation learning. In *Proceedings of the ieee/cvf International Conference on computer vision*, pp. 6391–6400, 2019.

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and Parikh, D. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

Guss, W. H., Houghton, B., Topin, N., Wang, P., Codel, C., Veloso, M., and Salakhutdinov, R. Minerl: A large-scale dataset of minecraft demonstrations. *arXiv preprint arXiv:1907.13440*, 2019.

Ha, D. and Schmidhuber, J. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Hafner, D. Benchmarking the spectrum of agent capabilities. *arXiv preprint arXiv:2109.06780*, 2021.

Johnson, J., Hariharan, B., Van Der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., and Girshick, R. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2901–2910, 2017.

Konar, A. *Artificial intelligence and soft computing: behavioral and cognitive modeling of the human brain*. CRC press, 2018.

Lai, Z., Zhang, H., Wu, W., Bai, H., Timofeev, A., Du, X., Gan, Z., Shan, J., Chuah, C.-N., Yang, Y., et al. From scarcity to efficiency: Improving clip training via visual-enriched captions. *arXiv preprint arXiv:2310.07699*, 2023.

Lake, B. M. and Baroni, M. Human-like systematic generalization through a meta-learning neural network. *Nature*, 623(7985):115–121, 2023.

Li, B., Ge, Y., Ge, Y., Wang, G., Wang, R., Zhang, R., and Shan, Y. Seed-bench-2: Benchmarking multimodal large language models. *arXiv preprint arXiv:2311.17092*, 2023a.

Li, J., Li, D., Savarese, S., and Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023b.

Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pp. 121–137. Springer, 2020.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Liška, A., Kruszewski, G., and Baroni, M. Memorize or generalize? searching for a compositional rnn in a haystack. *arXiv preprint arXiv:1802.06467*, 2018.

Liu, H., Li, C., Li, Y., and Lee, Y. J. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023a.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023b.

Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023c.

Loula, J., Baroni, M., and Lake, B. M. Rearranging the familiar: Testing compositional generalization in recurrent networks. *arXiv preprint arXiv:1807.07545*, 2018.

Majumdar, A., Ajay, A., Zhang, X., Putta, P., Yenamandra, S., Henaff, M., Silwal, S., Mcvay, P., Maksymets, O., Arnaud, S., et al. Openeqa: Embodied question answering in the era of foundation models. In *2nd Workshop on Mobile Manipulation and Embodied Intelligence at ICRA 2024*, 2024.

Molano-Mazon, M., Barbosa, J., Pastor-Ciurana, J., Fradera, M., Zhang, R.-Y., Forest, J., del Pozo Lerida, J., Ji-An, L., Cueva, C. J., de la Rocha, J., et al. Neurogym: An open resource for developing and sharing neuroscience tasks. 2022.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Rigotti, M., Barak, O., Warden, M. R., Wang, X.-J., Daw, N. D., Miller, E. K., and Fusi, S. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497 (7451):585–590, 2013.

Saikh, T., Ghosal, T., Mittal, A., Ekbal, A., and Bhattacharyya, P. Scienceqa: A novel resource for question answering on scholarly articles. *International Journal on Digital Libraries*, 23(3):289–301, 2022.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35: 25278–25294, 2022.

Shah, D., Osiński, B., Levine, S., et al. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Conference on robot learning*, pp. 492–504. PMLR, 2023.

Sun, W., Advani, M., Spruston, N., Saxe, A., and Fitzgerald, J. E. Organizing memories for generalization in complementary learning systems. *Nature neuroscience*, 26(8): 1438–1448, 2023.

Team, O. E. L., Stooke, A., Mahajan, A., Barros, C., Deck, C., Bauer, J., Sygnowski, J., Trebacz, M., Jaderberg, M., Mathieu, M., et al. Open-ended learning leads to generally capable agents. *arXiv preprint arXiv:2107.12808*, 2021.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Tsimpoukelli, M., Menick, J. L., Cabi, S., Eslami, S., Vinyals, O., and Hill, F. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021.

Wang, X., Wu, J., Chen, J., Li, L., Wang, Y.-F., and Wang, W. Y. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4581–4591, 2019.

Xu, P., Shao, W., Zhang, K., Gao, P., Liu, S., Lei, M., Meng, F., Huang, S., Qiao, Y., and Luo, P. Lvlm-ehub: A comprehensive evaluation benchmark for large vision-language models. *arXiv preprint arXiv:2306.09265*, 2023.

Yang, G. R., Ganichev, I., Wang, X.-J., Shlens, J., and Sussillo, D. A dataset and architecture for visual reasoning with a working memory. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 714–731, 2018.

Ye, Q., Xu, H., Xu, G., Ye, J., Yan, M., Zhou, Y., Wang, J., Hu, A., Shi, P., Shi, Y., et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023.

Yu, W., Yang, Z., Li, L., Wang, J., Lin, K., Liu, Z., Wang, X., and Wang, L. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.

Zhai, X., Wang, X., Mustafa, B., Steiner, A., Keysers, D., Kolesnikov, A., and Beyer, L. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18123–18133, 2022.

Zhang, P., Li, X., Hu, X., Yang, J., Zhang, L., Wang, L., Choi, Y., and Gao, J. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5579–5588, 2021.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Zhou, L., Xu, C., and Corso, J. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Zhu, D., Chen, J., Shen, X., Li, X., and Elhoseiny, M. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

A. Appendix

A.1. Related Works

A.1.1. Large Multimodal Models

Continual advancements in pretrained large-scale multimodal models are driving progress in a wide array of downstream tasks. Given the significant computational expense associated with end-to-end pretraining, it is common to utilize readily available pretrained vision models alongside Large Language Models (LLMs) such as (OPT (Zhang et al., 2022), FlanT5 (Chung et al., 2022), Vicuna (Chiang et al., 2023), LLaMA (Touvron et al., 2023)). Pioneering LMMs such as VisualGPT (Chen et al., 2022) and Frozen (Tsimpoukelli et al., 2021) have highlighted the advantages of leveraging pre-trained multimodal models. The primary challenge lies in achieving cross-modal alignment, given that LLMs typically lack exposure to images during their unimodal pretraining phase. LMM research is coalescing around a key strategy known as the *"visual instruction tuning"*, which involves a two-phase training process: firstly, a vision-language alignment pretraining stage, and secondly, a visual instruction tuning stage.

A range of methods and models have been developed to enhance the capabilities of LMMs. Early approaches uses a frozen object detector for visual feature extraction (Chen et al., 2020; Li et al., 2020; Zhang et al., 2021) while LiT (Zhai et al., 2022) borrowed a frozen pretrained image encoder from CLIP (Radford et al., 2021). More recently, Frozen (Tsimpoukelli et al., 2021) and Flamingo (Alayrac et al., 2022) have adopted an image-to-text generation approach, prompting the language model to generate text based on an input image. However, in BLIP-2 (Li et al., 2023b), it was shown that this approach is not adequate for overcoming the modality gap and instead proposed a Querying Transformer (Q-Former), acting as a visual resampler, and a two-stage bootstrapping pretraining method, which led to models that outperformed Flamingo80B (Alayrac et al., 2022) in zero-shot VQAv2 tasks with fewer trainable parameters. The Q-former architecture was also adopted by later work such as InstructBLIP (Dai et al., 2023) and Qwen-VL (Bai et al., 2023).

Moreover, GPT-4 (Achiam et al., 2023) has demonstrated remarkable proficiency in multi-modal dialogues with humans. Models like LLaVA (Liu et al., 2023b) and MiniGPT-4 (Zhu et al., 2023) have sought to emulate its performance by integrating a fully connected vision-language cross-modal connector, which significantly reduces the need for paired image-text data during pretraining. Both have shown notable proficiency in following natural instructions and visual reasoning. Beyond image-based LMMs, there have been developments in models that specialize in video information processing. PaLM-E (Driess et al., 2023) incorporates continuous real-world sensor data into LMMs, facilitating the unification of real-world perceptions with human language. Video ChatCaptioner (Chen et al., 2023a) leverages ChatGPT's conversational interface to enhance its understanding of video spatiotemporal contexts.

A.1.2. Multimodal benchmarks

The rapid progression of LMMs has driven the need for comprehensive benchmarks to assess their multifaceted capabilities. Traditionally, datasets primarily focus on computer vision tasks – classification, detection, segmentation, captioning, visual generation, and editing (Computer-Vision-in-the-Wild, 2024) – where instructions are implicitly integrated. However, these primarily unimodal datasets do not adequately test the models' proficiency in multimodal information alignment, revealing a limitation in our ability to fully evaluate the LMMs.

To address this, the community has turned to human-annotated datasets like MS-COCO (Lin et al., 2014) and web-scraped collections such as YFCC-100M (Goyal et al., 2019) to forge better-aligned multimodal datasets. Enhanced by additional data cleansing methods and supported by CLIP-like models, datasets like Conceptual 3M/12M (Lai et al., 2023) and LAION-5B (Schuhmann et al., 2022) have not only improved in descriptive quality and text-image alignment but also in scale. Despite these advancements, a crucial capacity of LMMs which is to effectively follow multimodal vision-language instructions, remains inadequately assessed by current benchmarks. To address this gap, several benchmarks have pivoted toward evaluating cognitive skills and systematic, quantitative assessments. Visual Question Answering (VQA) datasets like ScienceQA (Saikh et al., 2022) play a crucial role in examining LMMs' multimodal reasoning capabilities. More recent benchmarks such as MME (Fu et al., 2023) and LLaVA-Bench (Liu et al., 2023a) utilize manually crafted questions and answers for precise evaluations, while platforms like MMBench (Liu et al., 2023c) adopt ChatGPT-driven techniques for data creation and response generation. Notably, ShareGPT4V (Chen et al., 2023b) introduces a dataset with high-quality captions, initially derived from GPT4-Vision and subsequently expanded, reflecting a trend towards more sophisticated and scalable evaluation frameworks.

The exploration into video-based datasets expands the assessment scope to include temporal integration, covering areas like video captioning, event segmentation, and action prediction. Datasets from video game environments (e.g. Minecraft (Guss et al., 2019), XLand (Team et al., 2021), and Crafter (Hafner, 2021)) and Video Question Answering tasks (e.g. YouCook2 (Zhou et al., 2018)) are pivotal in evaluating models' strategic understanding and instructional adherence. Additionally, benchmarks like Seed-Bench-2 (Li et al., 2023a) and various perception tests pose further challenges for LMMs by testing their efficacy in navigating and interpreting complex, multimodal data streams. Yet, a gap persists in evaluating models' precision in following instructions across sequential images, a gap that the iWISDM environment aims to bridge.

A.2. iWISDM Construction Details

Inspired by prior work (Yang et al., 2018), iWISDM generates tasks through a 3-phase procedure: **(1)** Task graph construction; **(2)** Node initialization and; **(3)** Trial instantiation. Dividing the generation procedure into distinct phases eliminates the necessity of constructing the task graph anew for each task trial. Once all properties associated with the nodes in a given task graph have been specified (phase 2), the user can generate any number of trials of that particular task, each with potentially different stimuli, ground-truth actions, as well as any other inherent stochastic values such as the number of delay frames. In general, iWISDM creates tasks following:

$$\mathbf{f}, i, \mathbf{r} = \texttt{iWISDM}(G)$$

where, G denotes the task graph, **f** the sequence of visual frames, i the corresponding language instructions, and **r** the sequence of ground-truth actions for each visual frame within **f** according to the instruction i.

The task graph G can either be specified by the user or generated automatically via `AutoTask`. The major distinction between the two resides in the initial task graph construction phase (phase 1). In contrast to the user-specified mode, where the user needs to manually define the task graph, in `AutoTask` mode, the user needs to only specify the paramters listed in the following section.

A.2.1. Task graph construction

In iWISDM, tasks are instantiated as directed, acyclic, connected task graphs. Each node represents a predefined operator that specifies the logic of the task (we use the terms "node" and "operator" interchangeably). Task operators take downstream stimuli/actions as input and output stimuli/actions based on their definitions. All operators must have parent/upstream operators except, root operators that define the actions of a task. Collection of interconnected operators form minimal sub-graphs that define sub-tasks (e.g. `Get` operators are root operators that define the subtask: *What is the attribute of an object?*). Under user-defined connectivity constraints, sub-graphs can be combined to generate corresponding compositional tasks. The *depth* of the graph is measured by the longest path from the root operator to any other operator within the graph.

Each operator has a customizable set of rules that constrains its connections. The specific operators and their permissible connections are described below: (see Appendix A Figure 1 for visualization):

- **Functional Operators:** `Select`: This operator retrieves stimuli based on three criteria: i) "When": the frame on which a stimulus appears; ii) "Where": the location on the frame in which the stimulus appears, and; iii) "What": depending on the particular stimulus set, other details of the object that determines its identity. For example, when using the default ShapeNet stimulus set, the stimuli have three attributes: category (such as car versus plane), identity (which specific car), and view angle (the angle from which the stimulus is rendered). `Select` operators that have no downstream connections are the terminal nodes of the graph. Conversely, their potential downstream operators may include any `Get*` functional operator. `Switch`: Based on the output action of a boolean task, this operator connects the logic to one of two possible paths. Its compulsory downstream connection must be a boolean operator, while its typical upstream connections are subtasks graphs. `Get*`: This group of operators is responsible for fetching specific properties of stimuli, such as category, location, or identity, exemplified by operators like `GetCategory`, `GetLocation`, and `GetIdentity`. Its direct downstream connection is always a `Select` operator, and its upstream connections can be any boolean operator. `CONST`: The simplest form of operator, `CONST` represents a fixed value. It is often used as a downstream connection for boolean operators that compare attributes.

- **Boolean Operators:** `Exist`: Paired with a specific property value (e.g. "Desk"), this operator tests whether an object with the property value ("Desk") exists and generates a boolean output which can function as an action generator or as a downstream connection for `Switch` and other boolean operators. `And`, `IsSame`, `NotSame`, `Or`: These boolean operators process inputs from two boolean operators to produce a boolean outcome. They are critical for constructing logical conditions within the task graph.

Using these operators, iWISDM provides two possible modes of constructing task graphs: user-specified and automatic.

- **User-specified task graph construction.** In this mode, users have the freedom to fully specify the task graph by manually creating an instance of NetworkX directed graph. Doing so requires a manual definition of all nodes (operators) and edges (connection between operators) within a task graph, thereby establishing the desired task operation logic. Subsequently, the task graph can be fed to the generator function (`write_trial_instance`) to yield the corresponding task trials. One potential application of this mode is to replicate trials from a specific task such as the *n-back* or *contextual decision-making* tasks. In our core build of iWISDM, we have incorporated a collection of classic tasks from neuroscience and cognitive science literature that users can readily access.
- `AutoTask` **graph construction.** In `AutoTask` mode, users can define a custom *task space* with a set of hyperparameters to procedurally generate tasks. A task space delineates the complexity and permissible operations pertinent to task construction. The available hyperparameters include: **(1)** number of compositions, i.e. maximum number of `Switch` operators to compose subtasks **(2)** each task graph's maximum depth and maximum number of operators **(3)** set of operators to sample from.

In addition to above hyperparameters, in `AutoTask` mode, the allowed task structure is further constrained by the permitted connectivity for various operators (e.g. `And` operators must be followed by other boolean operators such as `IsSame`, `NotSame`, `And`, `Or`). A default operator connectivity is defined for all existing operators in our core build, but new connectivity rules could easily be added for any new user-defined operators. This is done through a Python dictionary, which details the allowed input and output operators for additional operator. By specifying these hyperparameters, iWISDM autonomously generates random runnable task graphs derived from the predefined task space. Each resultant task graph can then be used to generate specific trials.

To assure each task graph would comply with the connectivity rules between operators, we follow a backwards initialization process during `AutoTask` (Appendix A Figure.5). The task generation process starts from the root node and descends recursively. For each current node/operator n in the graph, its downstream operators C_n are randomly sampled based on the connectivity rules. As the graph depth approaches the specified maximum depth, the permissible operators with the shortest possible subtask depth are sampled into C_n. For instance in our core build, if n is the `And` operator, then only `IsSame`, `NotSame` are sampled since they have shorter subgraph depths than `And`, `Or`. Through this procedure, iWISDM `AutoTask` facilitates sampling from diverse task spaces with varying degrees of complexity specified by the user. The utilization of the connectivity rule dictionary and `Switch` operator guarantees the logical and feasible nature of the generated tasks, providing researchers with an extensive pool of tasks for investigation and exploration.

Together, iWISDM's two operating modes provide users with the flexibility to use the environment to either train or evaluate models on specific tasks (e.g. classic tasks from literature), as well as a wide variety of tasks adhering to specific guidelines as stipulated by the specified hyperparameters.

A.2.2. Node Initialization

The second step assigns values to each node within the task graph, thereby conferring logically coherent tasks. There are two critical challenges: the initialization of an independent task graph and the integration of multiple graphs in time.

A.2.3. Task trial instantiation

Regardless of the selected operation mode, for each task trial, iWISDM yields a frame sequence, an accompanying natural language instruction, and an action sequence. As delineated above, with graphs that define the task logic. The task graph then serves as the basis for instantiating task trials. Distractors and fixation cues are also added during this step. Each task trial comprises of the following distinct components:

- **Frame sequence**. The frame sequence consists of a array of images. They display visual information at each time step of the trial. Each frame is accompanied by a dictionary that contains the object properties within that frame. Images are stored as `PNG` files.

- **Natural language instructions**. Each trial is accompanied by a natural language instruction that explains the task steps and decision criteria. Natural language instructions are generated concurrently during task instantiation. A partial string is assigned to each operator in the task graph depending on its definitions and initialization (see Appendix Figure.6 for an example). This approach allows iWISDM to automatically produce contextually relevant natural language instructions that describe the task in a human-readable format for each task.

- **Action sequences**. Each trial also consists of an array of ground-truth actions at each frame of the trial. The action sequences could be used for supervised training and validation of the agents on the generated trials.

- **Distractors (optional).** To generate more attention-demanding tasks, users could opt-in for adding distractors to the visual frames. By specifying parameters during trial generation, distractors can be added without causing conflicts with existing task rules. We use the task-irrelevant stimulus properties to disambiguate the task-relevant stimuli from the distractors in the task instruction,. An example can be seen in Appendix Figure 9.

A.3. Main Features

At its core, iWISDM is designed to be scalable and extensible. To do so, we adopted a modular framework in which task rules are constructed compositionally by combining functional and boolean operators. The combination of these operators give rise to distinct tasks. Likewise, *task spaces* (i.e. collections of instantiable task graphs) are spanned by specifying the set of allowable operators and operator-operator connection rules (see Appendix A.2 for detailed definitions). We detail iWISDM's main features below:

Compositionality. Real world tasks are fundamentally compositional, as most tasks can be readily decomposed into sets of simpler subtasks that involve fewer sensory observations and cognitive operations. There are two crucial facets of compositionality that need to be considered: logical and temporal compositionality, both of which are common in daily human behavior, and allow individuals to efficiently handle complex tasks and adapt to dynamic environments. Cognitive processes involving task decomposition (i.e. breaking tasks into subtasks) and temporal combination of decision rules (i.e. combining the outcomes of subtasks) are fundamental to our ability to navigate the world around us.

Logical Compositionality: An agent's action can be viewed as a function of sensory observations, internalized world knowledge, prior actions, and its objectives (Ha & Schmidhuber, 2018). Yet, as outlined in the introduction, the decision-making process frequently decomposes into sub-decisions and information processing steps that are temporally constrained and required fewer observations. We define *logical compositionality* as how decisions can be combined hierarchically through boolean (i.e. `And`, `Or`, etc) and functional operators (i.e. `Switch` operator that asks for if...then...). As an example, consider the contextual decision making task (ctxDM, see Appendix Figure 4b or Figure 1d). In the task in Appendix Figure 1d, the subject needs to first compare the category of the objects in the first and third frames. If their categories are the same, then compare the category of the objects from the second and *third* frames. Otherwise, compare the category of the objects from the second and *fourth* frames. For this task, the task rule of the second task is conditioned upon the result of the first task, which makes a logical composition. As the depth and the total number of operators involved in the task grows, we can compose more complex logical structures.

Temporal Compositionality: Temporal compositionality is concerned with how different decision rules should be combined together to construct a complex task that extends *in time*. In real-world scenarios, individuals often face tasks that require multiple decisions to be made in sequence or in parallel. For instance, making coffee involves following decision rules to accomplish a sequence of tasks such as grinding the coffee beans, brewing, and pouring (Appendix Figure 4). While rule compositionality has been explored in several prior works (Lake & Baroni, 2023; Liška et al., 2018; Loula et al., 2018), the topic of temporal compositionality has attracted less attention in the field. This is potentially due to a lack of proper datasets or virtual environments that could enable such investigations. iWISDM is precisely engineered to fill this gap by enabling the generation of temporally compositional tasks made from combining simpler tasks in time (Appendix Figure 4).

Vast Task Space.Another important feature of iWISDM is the vastness of its task space. This feature naturally extends from the compositionality that is inherent to iWISDM's task generation procedure. The ability to produce a large number of distinct tasks is critically important for the robust evaluation of large multimodal models that are trained on increasing volume of information from the web.

Moreover, the vast number of instantiable tasks in iWISDM provides an opportunity for training or fine-tuning large multimodal models to improve their ability to follow instructions in a vision-language context.

Natural Language Task Instruction. Natural language provides a rich and convenient way of communicating complex information to biological or artificial agents. It has been shown that improvements on language understanding in LLMs directly enhances their generality (performing many tasks) and adaptation (0-shot generalization) (Brown et al., 2020; Radford et al., 2019). Due to their capacity to compress enormous knowledge bases, these models have been useful in various applications where traditionally human supervision had been necessary (Shah et al., 2023; Dasgupta et al., 2023). Perhaps for similar reasons, natural language input constitutes the core of most existing multimodal models and they are heavily trained on large text corpuses among others.

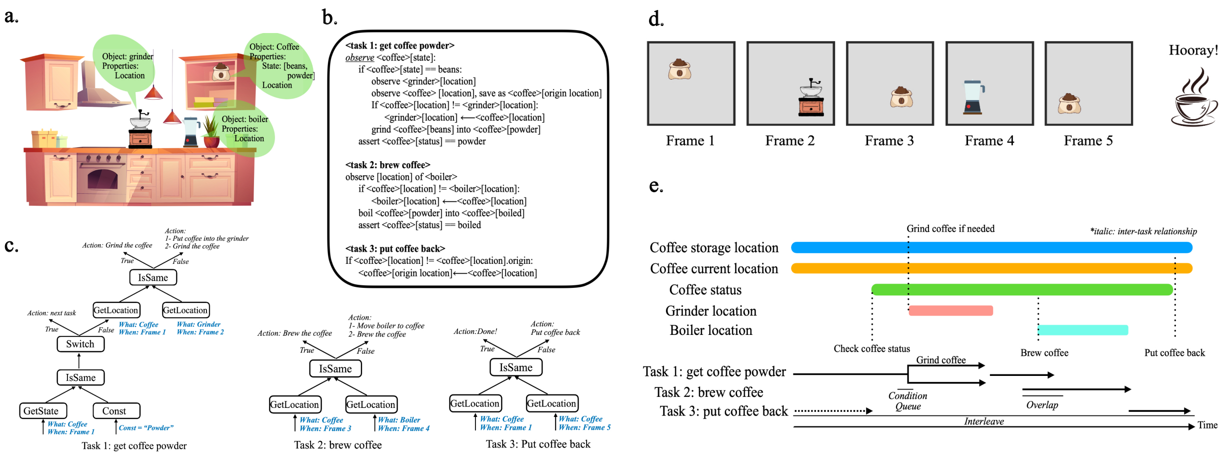
For this reason, in iWISDM, each task is accompanied by a simplified natural language instruction (See examples Appendix Figure5). When completing complex tasks, the instruction communicates first the task structure in terms of upcoming observations, then the task rules that determine the relationship between observations and the actions.

Automatic Task Generation. In contrast to prior virtual environments tailored for the purposes of cognitive or neuroscience investigations (Molano-Mazon et al., 2022), iWISDM can not only generate hand-crafted cognitive tasks such as classical decision making tasks (e.g. contextual decision making and n-back, Appendix Figure 1), but also allow procedural task generation from a pre-specified task space defined by a small set of hyperparameters (See Section 3.2 for details).

Customizability and Extensibility. We envision the future of iWISDM as a framework that will be continuously developed and expanded by the larger community of machine learning scientists, cognitive scientists, and neuroscientists. For this reason, we have designed iWISDM to be highly customizable in several ways.

- **Task operators**. iWISDM task rules are constructed from the interconnection of various building blocks called task operators. Task operators themselves are highly customizable and extensible, allowing users to define new task logic by inheriting the `Operator` class and overriding the `get_expected_input` function. Our core task operator set is adopted from a prior work (Yang et al., 2018) and includes `Get`, `IsSame`, `NotSame`, `And`, `Or`.
- **Visual inputs**. iWISDM allows any natural image set to be used as the stimulus set. In our core build, we use 2D projection of 3D object models from the ShapeNet dataset (Chang et al., 2015), where each stimuli is defined by a parameter vector consisting of `category`, `identity`, `pose angle`, `location`. We provide a template that allows users to seamlessly import alternative stimulus datasets.
- **Stimulus properties**. In addition to the visual stimuli themselves, users can define arbitrary object/stimuli properties to be used by the environment during task generation. In our core build of iWISDM, we use `category`, `identity`, `pose angle`, `location`. as the default object properties used during task generation. These properties are attached to each individual object in our stimulus set via an accompanying JSON file. Users can add custom properties by editing the accompanying JSON files for their stimuli of choice.

A.4. Additional Figures



```
<task 1: get coffee powder>
observe <coffee>[state]:
    if <coffee>[state] == beans:
        observe <grinder>[location]
        observe <coffee> [location], save as <coffee>[origin location]
        If <coffee>[location] != <grinder>[location]:
            <grinder>[location] ⟵<coffee>[location]
        grind <coffee>[beans] into <coffee>[powder]
    assert <coffee>[status] == powder

<task 2: brew coffee>
observe [location] of <boiler>
    if <coffee>[location] != <boiler>[location]:
        <boiler>[location] ⟵<coffee>[location]
    boil <coffee>[powder] into <coffee>[boiled]
    assert <coffee>[status] == boiled

<task 3: put coffee back>
If <coffee>[location] != <coffee>[location].origin:
    <coffee>[origin location]⟵<coffee>[location]
```

Figure 3. **Illustration of coffee making task as an example real world compositionally constructed task.** a) Cartoon depiction of a kitchen with typical objects within. The dialogue boxes highlight the relevant properties of the grinder, coffee bag, and boiler; b) Pseudocode detailing the coffee-making task, encompassing three subtasks – obtaining coffee powder, brewing the coffee, and returning the coffee; c) Graphical representation of the three subtasks as computational graphs. Blue text highlights the properties associated with each operator; d) An abstraction of the real-life coffee-making task as a task within the iWISDM environment. Frame sequences are generated using the task graphs illustrated in c; e) Graphic depiction of object properties actively in use (top) and subtasks (bottom). The colored bars in the upper panel signify the persistence of each object property during the execution of coffee-making task. The lower panel depicts the interrelations between the subtasks and various temporal compositional operations.

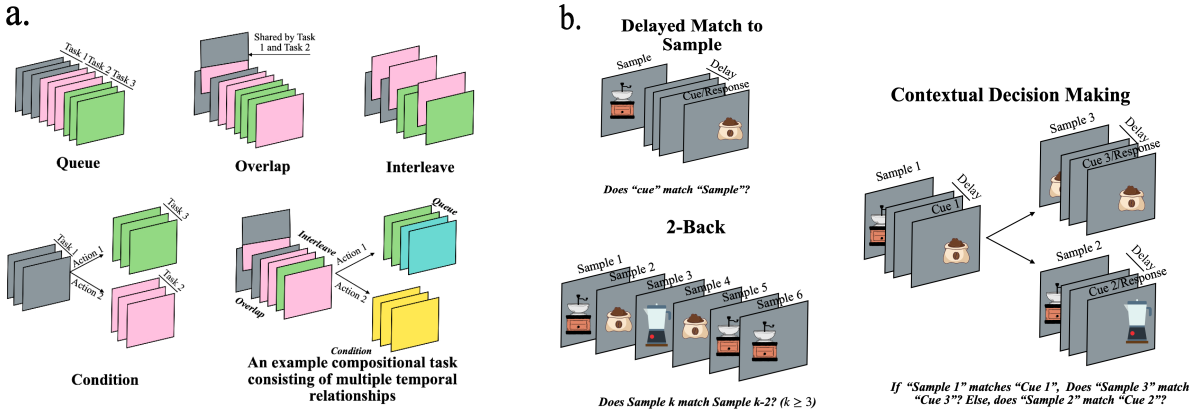
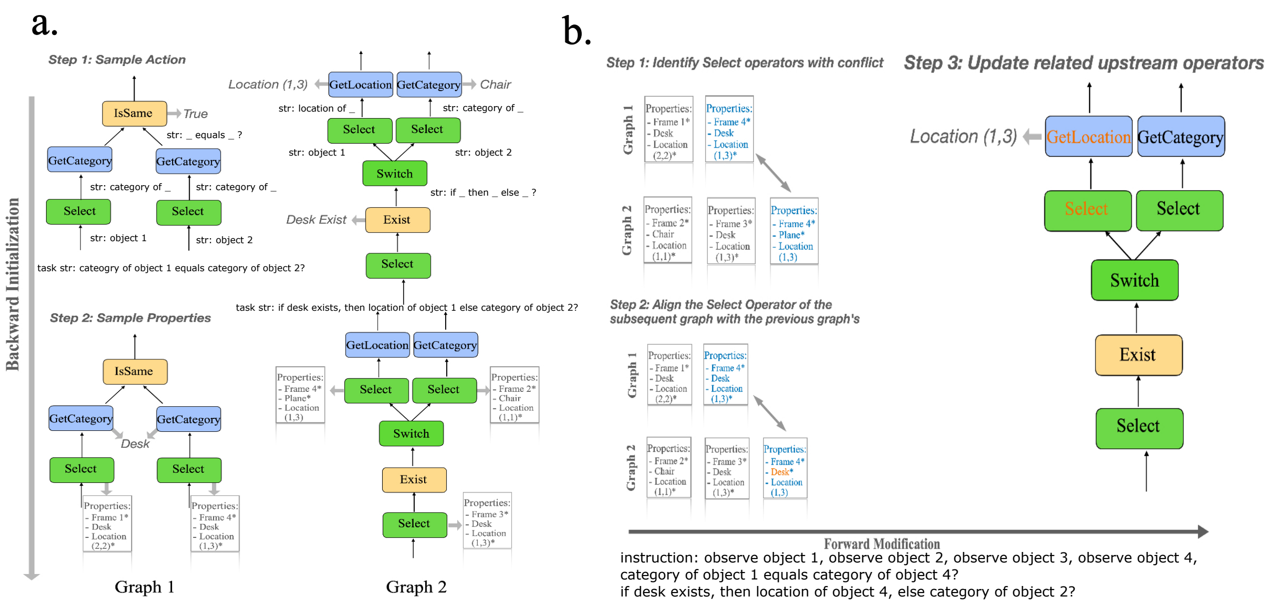


Figure 4. Illustration of various temporal compositional structures used in iWISDM and their application in instantiating classic decision making tasks. **a)** Four distinct temporal composition operations including *Queue*: where one task is completed before starting the next; *Overlap*: where two or more tasks share common information; *Interleave*: depicting the interwoven acquisition of information related to different tasks and; *Condition*: where the execution of a subsequent task depends on the outcome of the preceding one. An example compositional task consisting of multiple temporal relationships is shown. Frames are colored differently to highlight distinct tasks, with multicolored frames indicating shared information across tasks; **b)** Three classic cognitive tasks are exemplified: Delayed Match to Sample; 2-Back; and Contextual Decision Making.



instruction: observe object 1, observe object 2, observe object 3, observe object 4,
category of object 1 equals category of object 4?
if desk exists, then location of object 4, else category of object 2?

Figure 5. Task graph initialization and modification in iWISDM. This figure demonstrates the process iWISDM follows to initialize and modify a set of task graphs. We demonstrate the process using two examples: Graph 1 and Graph 2. **Backward Initialization (a)**: *Step 1*: Identify all operators linked to explicit (e.g.`IsSame` operator in Graph 1, operators in Graph 2) or implicit responses (where agents decide without a direct response, typically preceding the `Switch` operator, like the `Exist` operator in Graph 2), and assign responses to each of these operators, manually balancing output action. *Step 2*: Propagate properties to downstream operators. For example, if the response to `IsSame` is True, then the children `GetCategory` operators must have the same output. The `Select` operator usually receives one property from upstream operators, such as the category Desk in Graph 2's bottom `Select` operator. The "when" and "where" properties are randomly sampled and marked with asterisks. To illustrate instruction generation, the operator partial strings are shown as "str: ", and the blanks are filled by its children operators, following the direction of the arrows. However, during temporal composition, backward initialization can create conflicts between graphs, which leads **Forward modification phase (b)** to resolve these issues. *Step 1*: For each graph, gather properties from each `Select` operator and compare those with matching "when" to identify conflicts. For instance, a conflict is found in the "what" property at frame 4; Graph 1 `Select` assigns "Desk" while Graph 2's assigns "Plane" . Conflicts can also occur with the "where" property. Preference is given to modifying randomly assigned properties in subsequent graphs to minimize upstream modifications. *Step 2*: Adjust the "what" property in Graph 2 to "Desk", aligning with the Select operator in Graph 1. *Step 3*: Post-modification, it may be necessary to update actions or properties of upstream operators. In this scenario, tracing back to the `GetLocation` operator in Graph 2 shows that no changes are required for the "where" property, as it remains consistent with the `Select` operator's assignment.

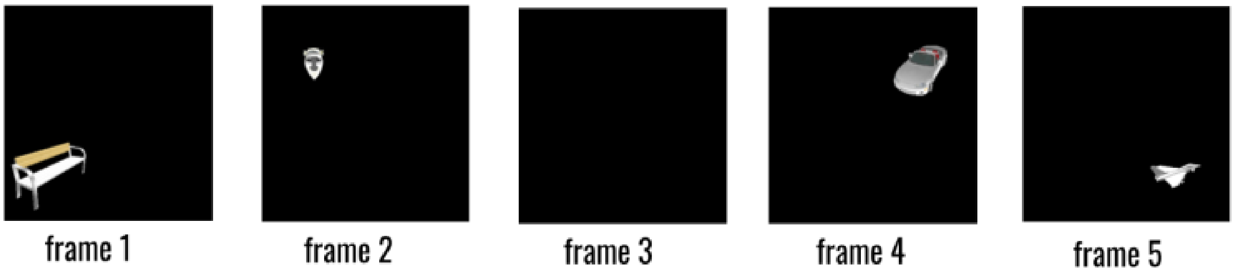


Figure 6. An example of low complexity task generated by iWISDM. Instruction: "observe object 1, observe object 2, delay, observe object 3, observe 4, location of object 3 equals location of object 2 or location of object 1 equals location of object 4?"

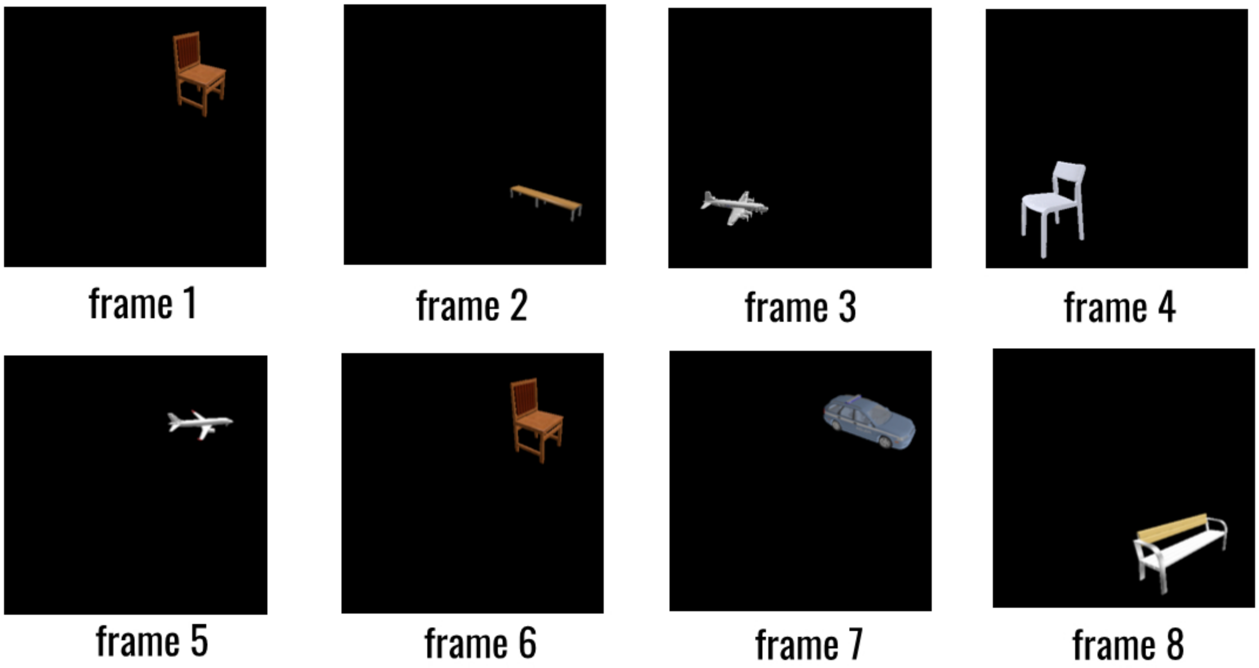


Figure 7. An example of medium complexity task generated by iWISDM. Instruction: "observe object 1, observe object 2, observe object 3, observe 4, observe object 5, observe object 6, observe object 7, observe 8, if location of object 7 equals location of object 2 or location of object 8 equals location of object 4, then location of object 3 equals location of object 1? else location of object 6 equals location of object 5?"

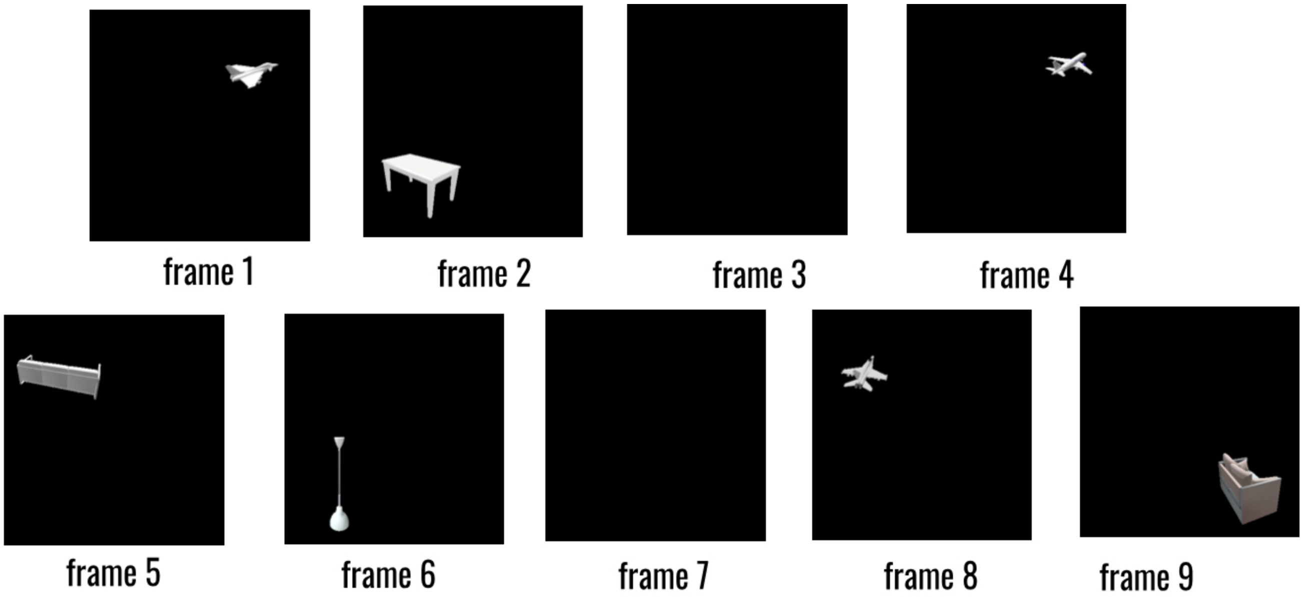


Figure 8. An example of high complexity task generated by iWISDM. Instruction: "observe object 1, observe object 2, delay, observe object 3, observe 4, observe object 5, delay, observe object 6, observe object 7, if location of object 3 not equals top right and location of object 2 equals location of object 5, then location of object 7 equals top left and location of object 6 equals location of object 4? else location of object 1?"

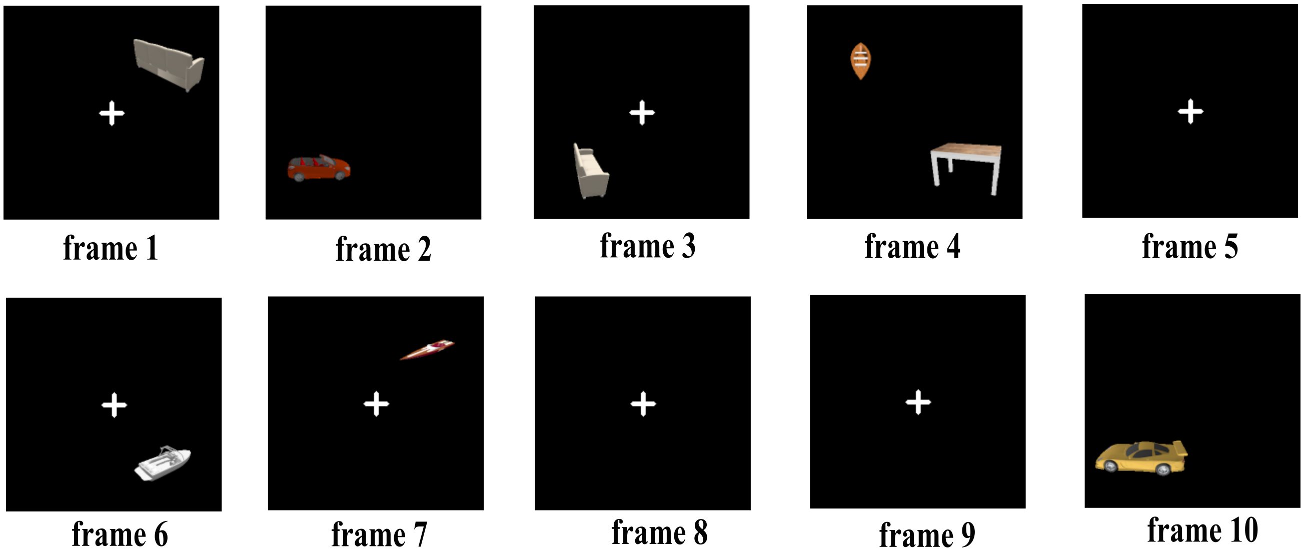


Figure 9. An example trial with distractors. Instruction: "observe object 1, observe object 2, category of object 2 not equals category of object 1? observe object 3,observe object 4 with location:, top left, category of object 4 equals couches or identity of object 3 equals identity of object 1? delay, observe object 5, observe object 6, delay, observe object 7, identity of object 7 equals identity of object 6 and identity of object 4 equals identity of object 3?"

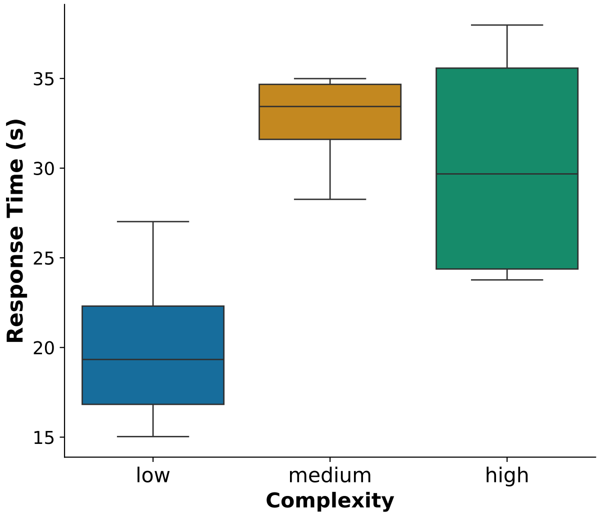


Figure 10. Human response times across complexity levels.

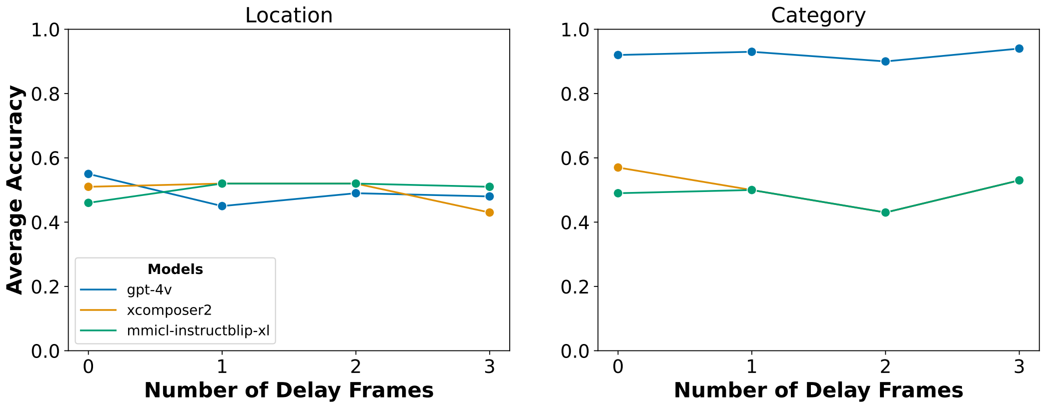


Figure 11. Average accuracy of GPT-4v, xcomposer2, and MMICL across a varying number of delay frames. Left is location-only features and right is category-only features .

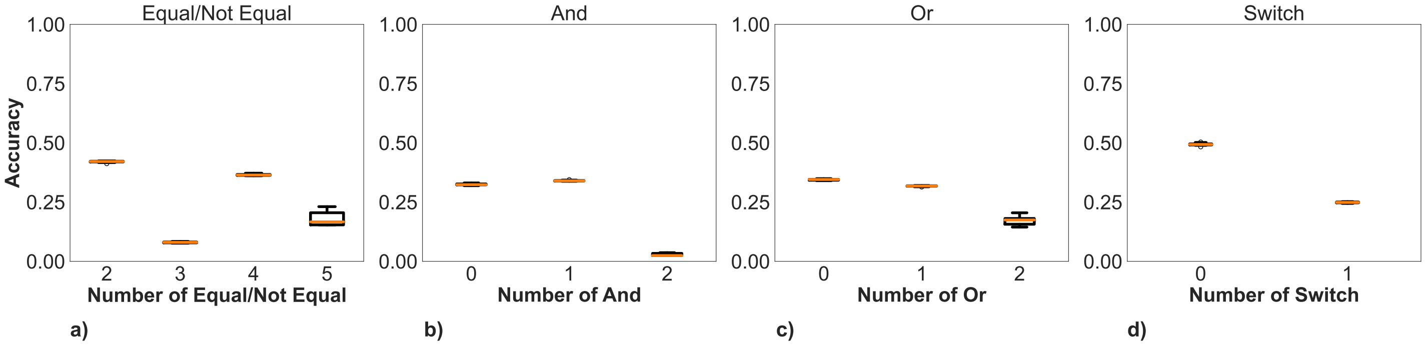


Figure 12. **a)** The performance accuracy of xcomposer2 and MMICL over the number of *Equal/Not Equal* boolean task operators. **b)** The performance accuracy of xcomposer2 and MMICL over the number of *And* boolean task operators. **c)** The performance accuracy of xcomposer2 and MMICL over the number of *Or* boolean task operators. **d)** The performance accuracy of xcomposer2 and MMICL over the number of *Switch* task operators.

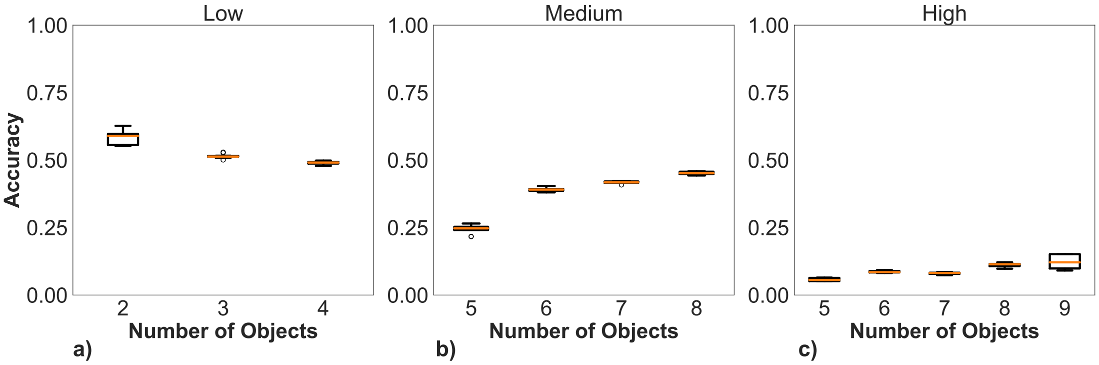


Figure 13. **a)** The performance accuracy of xcomposer2 and MMICL on low complexity tasks with varying numbers of object stimuli. **b)** The performance accuracy of xcomposer2 and MMICL on medium complexity tasks with varying numbers of object stimuli. **c)** The performance accuracy of xcomposer2 and MMICL on high complexity tasks with varying numbers of object stimuli.

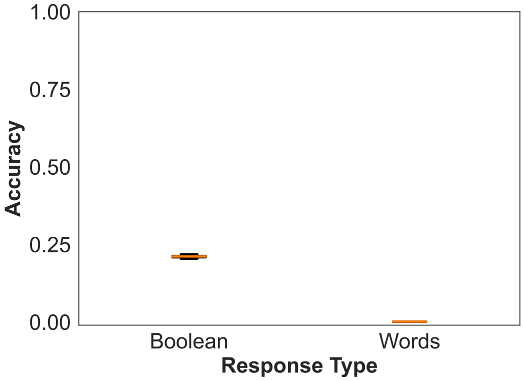


Figure 14. Accuracy of xcomposer2 and MMICL across response types for high complexity tasks.

A.5. Model evaluation prompts

A.5.1. GPT-4V, Claude-3, & Gemini-pro Low/Medium Complexity (all properties + examples included)

In this task we will show you a series of frame images. Each frame will either be blank (delay frame) or contain a 3D object. The objects within the task will ALWAYS be from one of 8 categories: benches, boats, cars, chairs, couches,lighting, planes, and tables. For each of these 8 categories there are 8 unique objects that could be used in the task. Any object which is sampled will be displayed as an image taken from a random viewing angle. The objects will be placed in one of four locations: top left, top right, bottom left, and bottom right.

A written instruction will be provided. Your goal is to follow the instructions and answer the question contained in the instruction. Answers will ALWAYS be one of the following: true, false.

Here is a simple example of the task ...

Task instruction: "observe object 1, observe object 2, location of object 1 not equal location: bottom left?"

Here are the corresponding frames ...

Answer: false. This is because the location of object 1 IS in the bottom left location.

Here is a simple example of the task...

Task instruction: "observe object 1, delay, observe object 2, category of object 1 equals category of object 2 ?"

Here are the corresponding frames ...

Answer: true. This is because the category of object 1 (lighting) IS equal to the category of object 2 (lighting).

Here is a simple example of the task...

Task instruction: "observe object 1, observe object 2, identity of object 1 equals identity of object 2 ?"

Here are the corresponding frames ...

Answer: true. This is because object 1 (a white table) IS identical to object 2 (the same white table).

Now please solve the following new task...

Task instruction: "observe object 1, observe object 2, delay, observe object 3, observe object 4, category of object 2 not equal category of object 3 or category of object 1 equals category of object 4 ?"

Here are the corresponding frames ...

What is the correct answer to this task? (respond EXACTLY and ONLY with one of the following answers: true, false). Provide your answer here:

A.5.2. GPT-4V Single-Frame Complexity (location property + no examples included)

In this task we will show you an image. Each image will contain a 3D object. The objects within the task will ALWAYS be from one of 8 categories: benches, boats, cars, chairs, couches,lighting, planes, and tables. For each of these 8 categories there are 8 unique objects that could be used in the task. Any object which is sampled will be displayed as an image taken from a random viewing angle. The object will be placed in one of four locations: top left, top right, bottom left, and bottom right.

A written instruction will be provided. Your goal is to follow the instructions and answer the question contained in the instruction. Answers will ALWAYS be one of the following: true, false.

Now please solve the following new task...

Task instruction: "observe object 1, category of object 1 not equals planes?"

Here are the corresponding frames ...

What is the correct answer to this task? (respond EXACTLY and ONLY with one of the following answers: true, false). Provide your answer here:

A.5.3. GPT-4V, Claude-3, & Gemini-pro High Complexity (all properties + examples included)

In this task we will show you a series of frame images. Each frame will either be blank (delay frame) or contain a 3D object. The objects within the task will ALWAYS be from one of 8 categories: benches, boats, cars, chairs, couches,lighting, planes, and tables. For each of these 8 categories there are 8 unique objects that could be used in the task. Any object which is sampled will be displayed as an image taken from a random viewing angle. The objects will be placed in one of four locations: top left, top right, bottom left, and bottom right.

A written instruction will be provided. Your goal is to follow the instructions and answer the question contained in the instruction. Answers will ALWAYS be one of the following: true, false, bottom right, bottom left, top left, top right, benches, boats, cars, chairs, couches, lighting, planes, tables .

Here is an example of the task...

Task instruction: "observe object 1, observe object 2, location of object 1 not equal location: bottom left ?"

Here are the corresponding frames ...

The correct answer: bottom right. This is because object 2 is located in the bottom right.

Here is a simple example of the task...

Task instruction: "observe object 1, delay, observe object 2, category of object 1 equals category of object 2 ?"

Here are the corresponding frames ...

Answer: lighting. This is because object 1 (a lamp) belongs to the category of lighting.

Here is a simple example of the task...

Task instruction: "observe object 1, observe object 2, identity of object 1 equals identity of object 2?"

Here are the corresponding frames ...

Answer: true. This is because object 1 (a white table) IS identical to object 2 (the same white table).

Now please solve the following new task...

Task instruction: "observe object 1, observe object 2, delay, observe object 3, observe object 4, observe object 5, if location of object 5 not equal location of object 2 , then location of object 1? else category of object 4 not equal tables or category of object 3 not equal couches?"

Here are the corresponding frames ...

What is the correct answer to this task? (respond EXACTLY and ONLY with one of the following answers: true, false, bottom right, bottom left, top left, top right, benches, boats, cars, chairs, couches, lighting, planes, tables). Provide your answer here:

A.5.4. INTERNLM-XCOMPOSER2 & MMICL LOW COMPLEXITY (ALL PROPERTIES INCLUDED + EXAMPLES EXCLUDED)

In this task we will show you a series of frame images. Each frame will either be blank (delay frame) or contain a 3D object. The objects within the task will ALWAYS be from one of 8 categories: benches, boats, cars, chairs, couches,lighting, planes, and tables. For each of these 8 categories there are 8 unique objects that could be used in the task. Any object which is sampled will be displayed as an image taken from a random viewing angle. The objects will be placed in one of four locations: top left, top right, bottom left, and bottom right.

A written instruction will be provided. Your goal is to follow the instructions and answer the question contained in the instruction. Answers will ALWAYS be one of the following: true, false .

Please solve the following task...

Task instruction: "observe object 1, delay, observe object 2, observe object 3, observe object 4, location of object 1 equals location of object 2 and category of object 3 equals category of object 4?"

Here are the corresponding frames ... <ImageHere> <ImageHere> <ImageHere> <ImageHere> <ImageHere> <ImageHere> What is the correct answer to this task? (respond EXACTLY and ONLY with one of the following answers: true, false). Provide your answer here:

A.5.5. INTERNLM-XCOMPOSER2 HIGH COMPLEXITY (ALL PROPERTIES INCLUDED + EXAMPLES EXCLUDED)

In this task we will show you a series of frame images. Each frame will either be blank (delay frame) or contain a 3D object. The objects within the task will ALWAYS be from one of 8 categories: benches, boats, cars, chairs, couches,lighting, planes, and tables. For each of these 8 categories there are 8 unique objects that could be used in the task. Any object which is sampled will be displayed as an image taken from a random viewing angle. The objects will be placed in one of four locations: top left, top right, bottom left, and bottom right.

A written instruction will be provided. Your goal is to follow the instructions and answer the question contained in the instruction. Answers will ALWAYS be one of the following: true, false, bottom right, bottom left, top left, top right, benches, boats, cars, chairs, couches, lighting, planes, tables .

Please solve the following task...

Task instruction: "observe object 1, observe object 2, observe object 3, observe object 4, delay, observe object 5, observe object 6, delay, observe object 7, if location of object 6 not equal location of object 2 or category of object 3 not equal category of object 4, then location of object 7 equals location of object 5? else category of object 1 ?"

Here are the corresponding frames ... <ImageHere> <ImageHere> <ImageHere> <ImageHere> <ImageHere> <ImageHere> <ImageHere> <ImageHere> <ImageHere> What is the correct answer to this task? (respond EXACTLY and ONLY with one of the following answers: true, false, bottom right, bottom left, top left, top right, benches, boats, cars, chairs, couches, lighting, planes, tables). Provide your answer here:

Table 1. Benchmark details for each level of complexity

Complexity	**# of allowed and/or operators in task**	**# of switch operators**	**# of trial frames**	**root operators**	**boolean operators**
Low	1	0	6	IsSame, And, Or, NotSame	IsSame, And, Or, NotSame
Medium	1	1	8	IsSame, And, Or, NotSame	IsSame, And, Or, NotSame
High	1-2	1	9	IsSame, And, Or, NotSame, GetLoc, GetCategory	IsSame, And, Or, NotSame