MATH TAKES TWO: A TEST FOR EMERGENT MATHEMATICAL REASONING IN COMMUNICATION

Michael Cooper & Sam Cooper
Cooper Cognitive
sam(@)coopercognitive.com
michael(@)coopercognitive.com

ABSTRACT

Although language models demonstrate remarkable proficiency on mathematical benchmarks, it remains unclear whether this reflects true mathematical reasoning or statistical pattern matching over learning formal syntax. Most existing evaluations rely on symbolic problems grounded in established mathematical conventions, limiting insight into the models' ability to construct abstract concepts from first principles. In this work, we propose Math Takes Two, a new benchmark designed to assess the emergence of mathematical reasoning through communication. Motivated by the hypothesis that mathematical cognition in humans co-evolved with the need for precise communication, our benchmark tests whether two agents, without prior mathematical knowledge, can develop a shared symbolic protocol to solve a visually grounded task where the use of a numerical system facilitates extrapolation. Unlike many current datasets, our benchmark eschews predefined mathematical language, instead requiring agents to discover latent structure and representations from scratch. Math Takes Two thus provides a novel lens through which to develop and evaluate models with emergent numerical reasoning capabilities.

1 INTRODUCTION

In recent years, neural network architectures, particularly large language models (LLMs), have achieved impressive performance in a wide array of tasks, including mathematical problem solving e.g. (Frieder et al., 2023; Lewkowycz et al., 2022). However, it is unlikely that these models enjoy the level of mathematical reasoning that we as humans do (Bengio and Malkin, 2024; Rudman et al., 2025). For instance, studies highlight how LLMs fail at both compositional reasoning and mathematical tasks beyond the level of more run-of-the-mill undergraduate problems (Frieder et al., 2023; Press et al., 2023; Petrov et al., 2025). These findings suggest that current architectures, while proficient in linguistic pattern completion, fall short of the kind of abstraction required for grounded mathematical reasoning, indicating the need for new approaches (Rudman et al., 2025).

Humans, by contrast, uniquely transformed intuitive physical insights into formal, predictive mathematical models. For example, while many animals possess an intuitive sense of gravity, Newton was the first to express it through precise mathematical laws. The earliest evidence of formal arithmetic is closely tied to the emergence of record-keeping in ancient Mesopotamia (Schmandt-Besserat, 1981). For example, Pythagorean triples appear in clay tablets used for land surveying, linking geometry to practical needs such as estimating crop yields (Mansfield, 2021). It is plausible that even before formal notation, the drive to communicate quantities quickly and accurately, e.g., "eight attackers are approaching", created selective pressure for symbolic reasoning. Symbolic language provides dramatic advantages over repetition, e.g., saying “attacker” eight times, or relying on shared physical context. From this perspective, mathematical reasoning may have co-evolved with symbolic communication as a compact means of encoding quantitative information.

Motivated by this view and the notion that mathematics emerges through compression and abstraction (Rissanen, 1978; MacKay, 2003; Bengio and Malkin, 2024), we propose a new benchmark to test whether artificial agents can develop basic mathematical reasoning through communication, without access to corpora containing human language. Specifically, we challenge models to invent and extrapolate discrete symbolic representations grounded in visual stimuli, without access to predefined

mathematical formalisms or human concepts of language. Our goal is being able to test the capacity of neural network architectures to rediscover counting including in $m \times n$ arrays (a potential route to multiplication) from the bottom up, in a communicative setting.

2 Related Work

Compositional language in communication: Several studies investigate compositional generalization from a language-focused perspective. Russin et al. (2019) introduce syntactic attention to improve generalization in sequence-to-sequence models on the SCAN benchmark (Lake and Baroni, 2018). Object-centric approaches also contribute to compositional generalization, especially in visual domains where disentangling viewpoint-invariant features remains a persistent challenge (Hinton et al., 2018; Locatello et al., 2020). More recently, benchmarks such as CoLA (Ray et al., 2023) evaluate alignment between compositional text and structured visual scenes. However, these studies typically assume pre-existing linguistic conventions and evaluate models within fixed symbolic systems. In contrast, our work draws inspiration from the idea that natural language did not emerge in isolation but as a tool for communication among agents. This shift in focus suggests that deeper insights into abstraction and generalization may arise from studying how structured symbolic communication can emerge interactively from grounded experience, rather than being imposed externally.

Emergent Language in Communication Settings: Key studies have also explored the notion that structured language emerges naturally through communication, especially in settings that mirror human language evolution. Experimental work by Verhoef and colleagues Verhoef (2012); Verhoef et al. (2016) shows that when humans interact under memory constraints or through repeated transmission, structured signaling systems tend to emerge. These studies highlight how limitations and social coordination pressures—such as bottlenecks, alignment, and co-adaptation—can give rise to compositional structure. Similar findings have emerged from neural simulations. For example, Lian et al. (2023) and Zhang et al. (2024) demonstrate that linguistic features like case marking and efficient word order arise only when communication is required. Broader frameworks, including NeLLCom-X (Lian and Andreas, 2024), and work by Kouwenhoven et al. (2022), reinforce this view by showing that language structure develops through cooperative interaction between agents grounded in shared perceptual environments. Our benchmark builds on these insights by encouraging agents to develop structured symbolic communication from scratch, under strict constraints on vocabulary and with strong generalization demands.

Emergent language in the "Bag-Select" game: Language emergence has often been studied in game-based scenarios, beginning with signaling games (Lewis, 1969). Neural implementations typically involve referential setups, where a sender describes a target image among distractors (Lazaridou et al., 2017), or generates symbol sequences based on a single image without context (Havrylov and Titov, 2017). More recent work explores numerical concepts, such as Zhou et al. (2024), who introduce a visual arithmetic task but reduce it to an image-to-text mapping. Our benchmark builds most directly on the "Bag-Select" game of Guo et al. (2020), where agents communicate object quantities to guide selection. That work finds that symbolic and visual inputs better support compositional language than linguistic inputs, measured by similarity to a reference grammar. We extend this by (i) grounding all input in 2D visual object arrays, (ii) using a fixed 8-token vocabulary, and (iii) evaluating communication success directly on compositional generalization tasks. Rather than comparing to a predefined grammar, we test whether agents can construct symbolic protocols that support extrapolation to novel quantities and symbols—a hallmark of emergent reasoning (Wiedemer et al., 2023). Math Takes Two builds on this tradition by explicitly grounding symbol emergence in visual reasoning and systematically evaluating out-of-distribution generalization.

3 Benchmarking Task

We introduce a benchmark inspired by a hypothetical early trading scenario: two agents must communicate about quantities of goods that are not physically present, relying solely on symbolic language grounded in shared visual understanding. This setting echoes the conditions under which mathematical reasoning may have first emerged, where language enabled the abstract representation and exchange of quantities without direct sensory input.

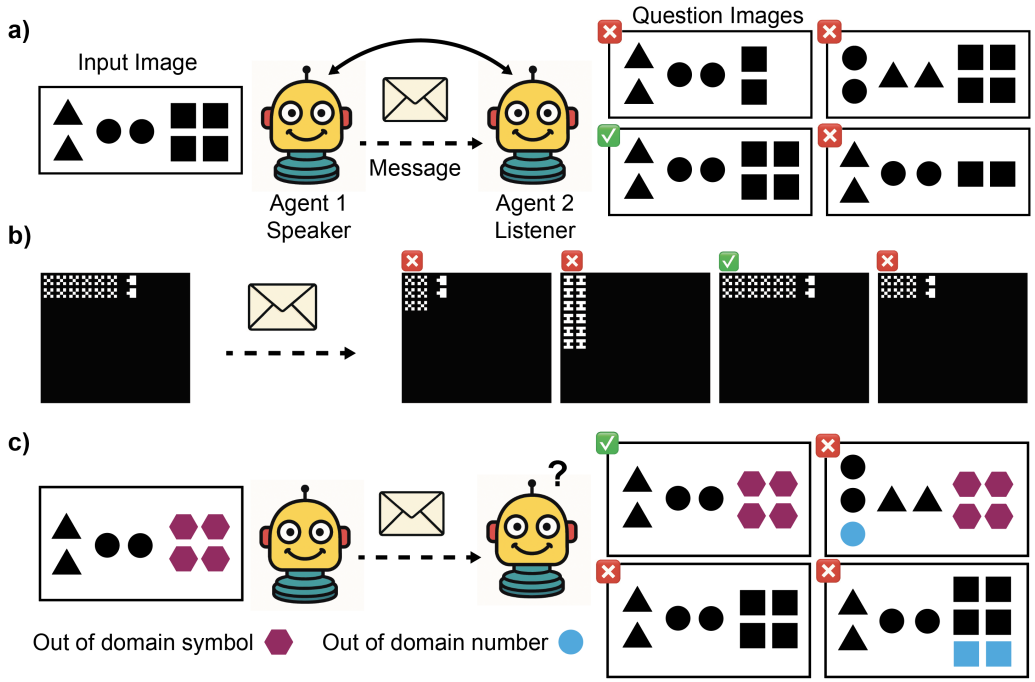

Figure 1: **Overview of the Math Takes Two benchmark. (a)** The *Speaker* receives an image depicting a collection of basic objects alone or in $m \times n$ arrays and communicates a symbolic string to the *Listener*, who must identify the correct target image from a candidate set. In the preconditioning phase agents may interact freely and communicate bidirectionally. **(b)** An example input image and questions set in the Math Takes Two benchmark. **(c)** During the practice and test phases, agents encounter examples with novel object types and numbers greater than previously seen. Only one candidate image is correct. Symbolic messages are the same length and vocabulary.

In our benchmark, a *Speaker* and a *Listener*, are trained to cooperate in the following protocol:

- The **Speaker** receives an input image representing a quantity of goods.
- The **Listener**, without access to the image, receives only a symbolic string from the Speaker and must select the correct image from a candidate set based on that string.

To limit complexity, the symbolic language is constrained to eight tokens [A, B, C, 0, 1, 2, +, *] selected to aid human performance at the task and understanding (we have already developed math from scratch) - but that should be treated uniformly by models.

This design also supports the generation of random images that can be described with strings of up to 8 characters. Crucially, the programmatic generation of these images gives researchers the option to design curricula that promote the emergence of structured communication. Example image–string pairs are shown in Figure 1, details on the language used to generate images is given in Appendix A.

While the language in Appendix A represents one valid encoding of image structure, we do not enforce any fixed syntax or parsing rules. Rather, the goal is to study whether models can discover useful internal representations purely from communication and visual grounding. The benchmark consists of three phases:

- **(a) Preconditioning Phase** Agents interact freely in a shared environment, where any number of images can be generated using a limited subset of objects and numeric values. This phase is intended to facilitate unsupervised discovery of mappings between visual

quantities and symbolic abstractions. Researchers are encouraged to design their own curricula during this stage.

- **(b) Practice Phase** Agents may proceed once sequentially through a set of 100 examples involving (i) novel object types and (ii) larger numerical values of objects than those seen in the preconditioning phase. Communication is restricted to a single symbolic string per image of at most 8 characters. The Speaker and Listener can further adapt using feedback on the response. Examples are presented in a logically ordered curriculum to facilitate deduction of new object types and numbers.
- **(c) Test Phase** Agents are evaluated on 100 new examples, again featuring unseen objects and quantities, but without feedback. The task requires extrapolation using one symbolic message per image, with a hard limit of 8 characters. Only one sequential forward pass is again permitted.

Constraints To ensure the benchmark reflects emergent vs. inbuilt reasoning, we disallow:

- Models with hardcoded mathematical capabilities.
- Models pre-trained on mathematical datasets.
- Models pre-trained on natural human language corpora.

We evaluate models based on their ability to develop grounded symbolic reasoning from scratch, echoing the hypothesized origins of early mathematical abstraction in human communication.

The Math Takes Two benchmark is available `here`. We also provide a dataset `here`.

4 Human Performance Evaluation

To benchmark model performance against human reasoning, we conducted a study involving 10 pairs of participants where at least one participant was required to have a technical background (undergrad engineering or sciences). Each pair was provided example images and tasked with developing a symbolic language using only the benchmark's predefined token set: `A, B, C, 0, 1, 2, +, *`. Symbolic strings were constrained to fewer than *8* characters (see Appendix B for further details).

Participants assumed fixed roles: the *Speaker* encoded each image as a symbolic string, while the *Listener* attempted to identify the corresponding image from a set of candidates using only that string. Participants were given notebooks to support the development of notation and strategy but were given no information about the image generation process or expected distribution shifts. Notebooks were submitted at the end of the exercise to evaluate the types of language systems developed by participants. The study comprised three phases:

- **(a) Learning Phase** Participants explored the environment freely, generating any number of images and developing their symbolic language collaboratively. They were also provided with a curated set of edge cases, designed to highlight the full space of possible shapes and quantities in the learning environment. No feedback or explicit instruction was given regarding underlying structure or distributional coverage.
- **(b) Practice Phase** Participants were separated and given 10 structured trials. In each trial, the Speaker viewed a new image and sent a symbolic string to the Listener, who then selected an image from a candidate set. Feedback was provided after each trial, allowing refinement of their shared language. The ordering of examples was designed to support progressive generalization.
- **(c) Test Phase** With no further feedback, participants completed 10 new trials involving previously unseen objects and quantities. This phase assessed whether the symbolic systems developed during the practice phase could support extrapolation under novel conditions. As in the model benchmark, only a single symbolic string could be communicated per trial.

The results of the 10 pairs of participants are summarized in Table 1. We include a description of the communication systems that pairs developed in Appendix B. Overall participants were adept at handling OOD tasks, and adapted their languages in real time to accommodate for previously unseen examples and numbers. Errors were split between specific failure modes of the developed languages, and simple errors of miscommunication including on in-domain questions.

Player	Overall Accuracy	OOD Symbol in Question	OOD Number in Question	OOD Symbol & Number in Q.
Human Prac.	**0.91**	*0.872*	*0.903*	*0.933*
Symb AE - F - Prac.	0.60	0.532	0.677	0.625
Symb AE - UF - Prac.	0.62	0.617	0.774	0.750
Symb Conv AE - F - Prac.	0.77	0.809	0.613	0.750
Symb Conv AE - UF - Prac.	0.83	0.851	0.742	0.813
VL Transformer - Prac.	0.83	0.809	0.774	0.750
Human Test	**0.87**	*0.844*	*0.731*	*0.692*
Symb AE - F - Test	0.60	0.511	0.538	0.308
Symb AE - UF - Test	0.57	0.444	0.423	0.231
Symb Conv AE - F - Test	0.66	0.578	0.385	0.615
Symb Conv AE - UF - Test	0.72	0.556	0.692	0.385
VL Transformer - Test	0.65	0.467	0.500	0.308

Table 1: **Model and human performance in the Math Takes Two benchmark.** Accuracy is reported for both human participants and symbolic autoencoder (AE) models, under frozen (F) and unfrozen (UF) training regimes, and a vision-language transformer (VL Transformer). OOD categories contain novel symbols, numbers, or both in the question images. OOD is a new symbol or number to the specific phase (i.e., OOD in the practice phase is new vs. preconditioning, OOD in the test phase is new vs. practice.)

5 Baseline performance of Symbolic Autoencoders

To establish a machine learning performance baseline, we evaluate two autoencoders that communicate via a *symbolic bottleneck* (Figure 2), similar to the networks proposed in Guo et al. (2020); Zhou et al. (2024), and Havrylov and Titov (2017). Each model is trained to reconstruct an input image through this bottleneck. A secondary similarity network is then trained to predict the correct answer from a set of four options, using the reconstructed image as input (Figure 2).

The symbolic bottleneck is implemented using a Gumbel-Softmax encoder (Maddison et al., 2017; Jang et al., 2017) that maps a flattened convolutional feature map (e.g., of shape $128 \times 5 \times 5$) into a fixed-length sequence of L discrete symbols, each drawn from a categorical vocabulary of size K. This symbolic sequence is then decoded by a learned decoder to produce a tensor matching the original feature map dimensions. The bottleneck and decoder are optimized jointly to minimize reconstruction loss, ensuring that the symbolic representation retains essential semantic information.

The two models differ in architectural depth:

- **Shallow Symbolic Autoencoder:** A single convolutional layer before symbolic encoding.
- **Deep Symbolic Autoencoder:** based on a fully convolutional autoencoder Masci et al. (2011), with symbolic encoding replacing the standard fully connected bottleneck after 5 warm up epochs.

Further architectural and training details are provided in Appendix C. We evaluate both models under two training regimes, as summarized in Table 1:

- *Frozen:* The symbolic autoencoder is pretrained on image reconstruction using a preconditioning dataset. Its weights are then frozen, and only the similarity network is trained on the question-answering task.
- *Unfrozen:* Both the symbolic autoencoder and the similarity network are trained end-to-end on the question-answering dataset.

Across all settings, model performance lags behind human accuracy, particularly on OOD questions. The gap is largest on examples featuring novel OOD shapes, indicating that while the symbolic bottlenecks support some generalization to new quantities and regions, they struggle to express or interpret unfamiliar visual primitives.

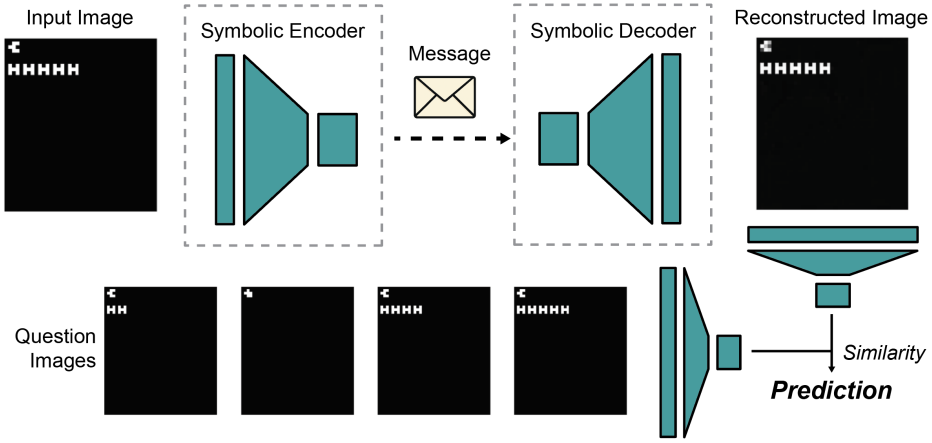

Figure 2: **Overview of the symbolic bottleneck model architecture.** The input image is first processed by a convolutional encoder that maps it to a latent feature representation. This representation is discretized into a symbolic message via a Gumbel-Softmax encoder. The message is then passed through a symbolic decoder to reconstruct the original image. For the question-answering task (bottom row), the reconstructed image is compared to a set of candidate question images using a similarity network, which produces a prediction based on the closest match.

Of note, we also find that while training the symbolic compression layer alongside the similarity network on the examples improves results on the practice questions, accuracy fell on the test dataset, indicating a reduction in generalization from explicitly learning to answer in-domain questions. These findings underscore the limitations of current symbolic compression schemes and motivate the development of models that can learn more generalizable communication protocols for visually grounded reasoning tasks.

6 LIMITATIONS

We present *Math Takes Two*, a novel benchmark designed to evaluate whether two neural agents can develop emergent reasoning capabilities through communication, and whether such capabilities generalize to OOD tasks. We note three key limitations of the benchmark below.

- **Shortcut-based OOD detection:** Models with strong capabilities for distinguishing in-distribution from OOD examples may achieve high performance without developing meaningful symbolic abstractions. To mitigate this, we designed specific questions that require not just identifying OOD inputs, but also selecting correct answers based on symbolic communication. Nevertheless, we cannot guarantee that language adaptation is strictly necessary for perfect accuracy.
- **Lack of explicit compositional evaluation:** The benchmark does not directly assess whether agents develop internal parsing mechanisms or compositional reasoning. Instead, we use performance on OOD tasks as an indirect proxy for abstraction. Naive solutions—such as hard-coded pixel-level matching across image grids—may still perform well without discovering reusable or generalizable symbolic structures. Benchmark integrity may be compromised if models exploit low-level heuristics or encode external knowledge specific to the test distribution.
- **Avoidance of pre-trained LLMs:** We also choose not to directly assess the performance of a pre-trained LLM, nor to employ an LLM teacher–student framework. This decision is intentional: the goal of the benchmark is to emphasize that mathematics was developed or discovered by humans in the absence of prior formal structures, likely emerging from evolved survival-driven cognitive biases. The core value of this benchmark, therefore, lies in understanding how analogous developmental objectives can be embedded within neural networks to enable genuine from-scratch learning.

7 FUTURE WORK

Our benchmark builds upon the "bag-selection" game proposed by Guo et al. (2020), extending it into a setting that tests both the emergence of structured symbolic communication and the capacity to generalize to examples with non-overlapping support—a gold standard for evaluating reasoning (Wiedemer et al., 2023). In doing so, we aim to encourage the development of models that can describe complex visual scenes, including features related to counting, orientation, and spatial arrays.

We now know that scaling of language models to vast parameter numbers leads to dramatic improvements in performance in language and arguably reasoning (Vaswani et al., 2017). Creating pretraining environments and scenarios that would allow large language models to develop languages in the absence of human formalisms, for example, as in Lian and Andreas (2024); Kouwenhoven et al. (2022), would be a key next step in trying to improve performance at this benchmark. Three behaviors observed in human players suggest further directions for improving model performance:

- **Incorporating external memory mechanisms:** Participants consistently used notebooks to externalize their symbolic protocols during both the development and testing phases. This aligns with observations by Bengio and Malkin (2024), who argue that external memory can compensate for limited working memory and enable more advanced reasoning. These findings suggest that models equipped with constrained working memory and access to symbolic registers or external memory modules may be better suited for this benchmark.
- **Modeling uncertainty and anticipating novelty:** Several participant pairs, without explicit instruction, developed fallback strategies or codes in anticipation of novel, previously unseen examples. This proactive behavior suggests that agents capable of modeling distributional uncertainty or maintaining flexible priors over possible test conditions may generalize more effectively in open-ended tasks.
- **Limitations on task breadth:** The current version of the benchmark is restricted to counting tasks over grid-based object arrays using an 8-token vocabulary. While this design isolates foundational symbolic reasoning under communication constraints, it does not yet explore more complex operations or larger vocabularies (e.g., 10 or 12 tokens). Extending the benchmark to include richer symbolic spaces and a broader range of reasoning tasks will also be an important avenue for future work.

8 CONCLUSIONS

Overall, *Math Takes Two* provides a testbed for evaluating whether neural agents can move beyond statistical pattern matching to acquire structured, symbolic communication in a grounded and generalizable manner. Our framework builds on the communicative bottleneck tradition of the Bag-Select game (Guo et al., 2020) and related emergent language research (Lazaridou et al., 2017), but extends these foundations with systematic out-of-distribution (OOD) tests and visual reasoning tasks that require symbolic extrapolation. Future iterations of this framework could incorporate elementary arithmetic operations. For instance, requiring agents to match a $bags \times objects$ input to a corresponding product in an output image. Such extensions would transform the benchmark into a complex image-to-image reasoning task, building on the symbolic counting challenges posed by Zhou et al. (2024) while enforcing OOD generalization in environments devoid of predefined symbolic scaffolding.

The absence of symbolic number systems in certain human cultures provides a compelling anthropological parallel to our experimental setup. Research on isolated Amazonian communities, such as the Pirahã and Mundurukú, suggests that without a linguistic "scaffolding" for exact numbers, humans rely on an Approximate Number System (ANS) to distinguish quantities like "one," "a few," or "many" (Pica et al., 2004; Gordon, 2004). These findings imply that while the capacity for logic may be innate, the transition from approximate perception to exact mathematical reasoning is fundamentally mediated by the development of a symbolic protocol. We hypothesize that in settings where the ecological context does not demand precise quantification, qualitative reasoning remains sufficient, and discrete symbolic concepts may never emerge. By stripping away predefined human tokens, *Math Takes Two* places neural agents in a similar "pre-symbolic" state, allowing us to observe whether the functional pressure of a communicative task is sufficient to drive the shift from approximate pattern matching to exact, symbolic representation.

More broadly, our work aligns with efforts to study emergent reasoning from first principles, such as the *DreamCoder* agent (Ellis et al., 2023; 2021), which combines symbolic program induction with visual and linguistic reasoning. Inspired by this line of research, we postulate that future environments could explore even richer mathematical topographies. This might include requiring agents to negotiate protocols for communicating geometric properties like area or angle using limited auxiliary tools, such as unit-length ropes. Success in such tasks would reflect the emergence of higher-order abstractions akin to geometry and the Pythagorean theorem. Ultimately, this path leads toward machine learning systems capable of not merely executing and extending human mathematical concepts, but inventing and reasoning about novel mathematical concepts in an open-ended manner built from the ground up.

Ethics Statement. Participants were recruited to take part in the study in pairs and provided informed consent for their anonymized results to be used for research purposes. No personally identifying information was collected or stored. The study involved a low-risk, educational task with no deception or intervention, and was therefore considered exempt from formal institutional ethics review.

This work investigates the emergence of symbolic reasoning in artificial agents. While advances in such capabilities may contribute to scientific understanding and the development of more general learning systems, they may also have broader societal implications, including impacts on labor and the deployment of increasingly autonomous systems. As with all research in this area, careful consideration of downstream uses and responsible deployment practices will be important as these technologies mature.

REFERENCES

Yoshua Bengio and Nikolay Malkin. Machine learning and information theory concepts towards an ai mathematician. *Bulletin of the American Mathematical Society*, 61(3):457–469, 2024.

Kevin Ellis, Catherine Wong, Maxwell Nye, Mathias Sablé-Meyer, Lucas Morales, Luke Hewitt, Luc Cary, Armando Solar-Lezama, and Joshua B Tenenbaum. DreamCoder: bootstrapping inductive program synthesis with wake-sleep library learning. In *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, New York, NY, USA, June 2021. ACM.

Kevin Ellis, Lionel Wong, Maxwell Nye, Mathias Sable-Meyer, Luc Cary, Lore Anaya Pozo, Luke Hewitt, Armando Solar-Lezama, and Joshua B Tenenbaum. Dreamcoder: growing generalizable, interpretable knowledge with wake–sleep bayesian program learning. *Philosophical Transactions of the Royal Society A*, 381(2251):20220050, 2023.

Simon Frieder, Luca Pinchetti, Alexis Chevalier, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Christian Petersen, and Julius Berner. Mathematical capabilities of ChatGPT. In *Advances in Neural Information Processing Systems*, volume 36, pages 27699–27744, 2023.

Peter C. Gordon. Numerical cognition without words: Evidence from amazonia. *Science*, 306(5695): 496–499, 2004. doi: 10.1126/science.1094492.

Shangmin Guo, Yi Ren, Serhii Havrylov, Stella Frank, Ivan Titov, and Kenny Smith. The emergence of compositional languages for numeric concepts through iterated learning in neural agents. In *The Evolution of Language: Proceedings of the 13th International Conference (EvoLang13)*. Evolang, 2020.

Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. *Neural Inf Process Syst*, abs/1705.11192, February 2017.

Geoffrey E. Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. In *International Conference on Learning Representations (ICLR)*, 2018.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations (ICLR)*, 2017.

Daniel Kouwenhoven, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Emergence of linguistic structure in cooperative referential games. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2873–2882. PMLR, 2018.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. In *International Conference on Learning Representations (ICLR)*, 2017.

D Lewis. Convention: A philosophical study. 1969.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, H Michalewski, V Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. *Neural Inf Process Syst*, abs/2206.14858:3843–3857, June 2022.

Ruihan Lian and Jacob Andreas. Nellcom-x: Emergent language learning through cooperative multi-agent communication. In *Proceedings of the 28th Conference on Computational Natural Language Learning*, 2024.

Ruihan Lian, Jacob Andreas, and Luke Zettlemoyer. Emergent case marking in neural agents through communicative pressure. *Transactions of the Association for Computational Linguistics*, 2023.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Olivier Bachem, David Tan, Jack Rae, Pushmeet Kohli, and Matthew Botvinick. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

David J C MacKay. *Information theory, inference and learning algorithms*. Cambridge University Press, Cambridge, England, 2003.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations (ICLR)*, 2017.

Daniel F Mansfield. Plimpton 322: A study of rectangles. *Found. Sci.*, 26(4):977–1005, December 2021.

Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Lecture Notes in Computer Science*, Lecture notes in computer science, pages 52–59. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

Ivo Petrov, Jasper Dekoninck, Lyuben Baltadzhiev, Maria Drencheva, Kristian Minchev, Mislav Balunović, Nikola Jovanović, and Martin Vechev. Proof or bluff? evaluating LLMs on 2025 USA math olympiad. *arXiv [cs.CL]*, March 2025.

Pierre Pica, Cathy Lemer, Véronique Izard, and Stanislas Dehaene. Exact and approximate arithmetic in an amazonian indigene group. *Science*, 306(5695):499–503, 2004. doi: 10.1126/science.1102085.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5687–5711, 2023.

Arijit Ray, Filip Radenovic, Abhimanyu Dubey, Bryan A Plummer, Ranjay Krishna, and Kate Saenko. Cola: A benchmark for compositional text-to-image retrieval. *Neural Inf Process Syst*, 36:46433–46445, May 2023.

J Rissanen. Modeling by shortest data description. *Automatica (Oxf.)*, 14(5):465–471, September 1978.

William Rudman, Michal Golovanevsky, Amir Bar, Vedant Palit, Yann LeCun, Carsten Eickhoff, and Ritambhara Singh. Forgotten polygons: Multimodal large language models are shape-blind. *arXiv [cs.CV]*, February 2025.

Jake Russin, Jason Jo, Randall C O'Reilly, and Yoshua Bengio. Compositional generalization in a deep seq2seq model by separating syntax and semantics. In *Proceedings of the 2019 Workshop on Cognitive Modeling and Computational Linguistics*, pages 52–58, 2019. doi: 10.18653/v1/W19-2907.

Denise Schmandt-Besserat. From tokens to tablets: A re-evaluation of the so-called "numerical tablets". *Visible language*, 15(4):321–344, 1981.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Tessa Verhoef. *The Origins of Duality of Patterning in Artificial Whistled Languages*. PhD thesis, University of Amsterdam, 2012.

Tessa Verhoef, Simon Kirby, and Bart de Boer. Emergence of combinatorial structure and economy through iterated learning with continuous signals. *Journal of Phonetics*, 54:57–68, 2016.

Thaddäus Wiedemer, Prasanna Mayilvahanan, Matthias Bethge, and Wieland Brendel. Compositional generalization from first principles. In A Oh, T Naumann, A Globerson, K Saenko, M Hardt, and S Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 6941–6960. Curran Associates, Inc., 2023.

Yichi Zhang, Ruihan Lian, and Jacob Andreas. Word order emergence in neural agents requires communication. In *International Conference on Learning Representations (ICLR)*, 2024.

Enshuai Zhou, Yifan Hao, Rui Zhang, Yuxuan Guo, Zidong Du, Xishan Zhang, Xinkai Song, Chao Wang, Xuehai Zhou, Jiaming Guo, Qi Yi, Shaohui Peng, Di Huang, Ruizhi Chen, Qi Guo, and Yunji Chen. Emergent communication for numerical concepts generalization. *Proc. Conf. AAAI Artif. Intell.*, 38(16):17609–17617, March 2024.

A SPECIFICS OF THE LANGUAGE USED TO DEVELOP THE ENVIRONMENT

Warning: this section contains spoilers as to how to encode the images. Readers may first enjoy attempting the task as described on the github page.

https://github.com/socooper/mathtakestwo/tree/main/player_env

Symbolic Shape Language. We define a compact symbolic language for generating and rendering structured shape-based scenes. Each program string represents a composition of primitive shape placement commands that are parsed and visualized by a canvas-based rendering engine. Programs are syntactically valid sequences over a vocabulary of predefined shape codes and trinary-encoded layout arguments. The total alphabet consists of 8 characters, yet the maximum number of shapes encountered, and number of shape possibilities is specifically chosen to exceed 8 to challenge players to make decisions at some point as to how to use combinations of characters to represent shapes or numbers, especially out of distribution. It is also noted that some shapes are only encoded with a single letter. These appear more frequently in the examples, though this behavior would be impossible to identify based on the number of examples given to human participants, thus perfect rediscovery of the underlying code for human players is likely impossible, and 'lossy compression' solutions are expected.

- **Vocabulary:**
  - Shape symbols: `A`, `B`, `C`, and all two-letter combinations from this set (e.g., `AB`, `CC`), for a total of 12 shape types.
  - Numeric arguments: strings over the digits `0`, `1`, and `2`, interpreted as base-3 (trinary) integers (e.g., `11` = 4).

  - Special operators: $\star$ horizontal or grid layout mode, + program concatenation (next shape command on a new region)
- **Program Syntax:** Each symbolic program is a string composed of one or more layout commands concatenated with +. Each command is parsed into one of four rendering primitives:
  - `SHAPEcc*cc`: Grid layout with `rows = parse_trinary(cc), cols = parse_trinary(cc)`
  - `SHAPE*cc`: Horizontal row of shapes (1 row, multiple columns)
  - `SHAPEcc`: Vertical column of shapes (1 column, multiple rows)
  - `SHAPE`: Single shape at origin

  For example:
  - `B12*12` $\rightarrow$ Draw shape `B` in a 5×5 grid
  - `BB11+AB2` $\rightarrow$ Draw `BB` in a vertical stack of 4, then draw `AB` in a vertical stack of 2 in a separate region
  - `C` $\rightarrow$ Draw a single `C` shape
- **Program Generation:** Programs are generated by a Markov model with token transitions defined over a set of states including shapes, numbers, and special operators.
  - Transitions from shape tokens yield either number tokens, operator tokens ($\star$, +), or `END`.
  - Number tokens can be followed by layout operators or termination.
  - Operator $\star$ initiates array layout and is followed by a numeric width.
  - Operator + resets the parser state and begins a new shape command.

  The generator samples valid sequences of up to 8 tokens using this transition model.

This symbolic language allows the concise generation of richly structured visual inputs, enabling studies of compositional generalization and discrete communication in neural agents.

B HUMAN SURVEY DETAILS AND STRATEGIES DEVELOPED

Participants engaged via a set of printout pdfs provided online at https://github.com/socooper/mathtakestwo/tree/main/player_env. Participants received the `examples_1`, `examples_2`, and `edge_cases` printouts. Each pair was instructed to develop a symbolic language allowing the speaker to describe the example image and the listener to select one of four answer options. During the learning phase, participants could take notes freely and were informed that in the practice phase, they would be separated. Communication would then be restricted to: (1) The speaker sending an 8-character message, (2) The listener responding with a selection, and (3) The speaker confirming correctness. In the test phase, only message passing was permitted, with results recorded afterward.

Messages could be shorter than 8 characters but could not include spaces (to reduce writing burden; padding with symbols such as $\star$ was possible but rarely used). Pairs typically spent 30–90 minutes developing their code, and approximately 10 minutes completing the practice and test phases. Performance data was recorded in both phases, along with observations of the communication strategies adopted. A summary of the three strategy types is provided below.

PLAYER ENCODING STRATEGIES

Participants typically adopted one of three distinct strategies when encoding images into 8-character symbolic messages drawn from an 8-symbol alphabet.

- **(a) Decision Tree Encoding:** Some participants employed a *decision tree* strategy, where each character in the message corresponded to the answer to a categorical question about the image—for example, "What is the symbol of the most common element?" or "What is the greatest number of elements in any one direction?" Typically, two or three such questions were formulated, with each assigned to a dedicated slot in the message or separated using a

spacer. All eight symbols were often used to represent predefined answers. This approach enabled participants to correctly answer many queries, including out-of-distribution (OOD) examples, due to redundancy in the encoding. For instance, even if the symbol itself was novel, the associated number might remain within the training distribution.

- **(b) Index-Based Encoding:** Most groups adopted an *index-based* strategy, in which specific positions within the 8-character message were reserved for key attributes—typically the symbol type, row number, and column number of the first two shape groups. This lookup-style method allowed for rapid decoding and quick identification of the correct image by visual inspection. However, it proved brittle under out-of-distribution conditions, such as when the number of shapes or shape types exceeded the 8-character limit. In such cases, additional information had to be omitted or compressed, reducing the method's robustness.
- **(c) Programmatic Encoding:** Two participant pairs employed a *programmatic* strategy, aiming to reverse-engineer the underlying rules believed to govern image generation. One group adopted a base-6 naming convention for shapes and numbers, using all alphanumeric characters, with "+" to separate groups and "*" to denote operations such as shape $\times$ row $\times$ column. Another group independently rediscovered the base-3 (ternary) numerical system used in image construction and used the 8-character message to first encode shape identity, followed by ternary digits representing the total number of shapes. Although this latter approach failed to capture orientation differences, resulting in one incorrect answer during practice, it achieved high compression and led to perfect accuracy on OOD test questions, aside from a single human error on an in-distribution example.

We also recorded individual participant scores to ensure that results were reproducible across groups. These are provided on the github page in the human results table. Across the 10 player pairs, the practice mean and standard deviation were $\mu = 9.1, \sigma = 0.81$, and test mean and standard deviation were $\mu = 8.7, \sigma = 0.56$. We considered this low standard deviation to mean ratio a sign that test results were reproducible across the set of player pairs assessed.

C Model Architecture and Hyperparameters

Symbolic Autoencoder This model serves as our simplest baseline, encoding images directly into discrete tokens.

- **Symbol Encoder:** Architecture consists of `Dropout2d` $\rightarrow$ `Conv2d`(1, 64, 3) + ReLU $\rightarrow$ `AdaptiveAvgPool2d`((1, L)) $\rightarrow$ `Conv1d`(64, K, 1). It utilizes Gumbel-softmax sampling ($\tau = 0.5$) to produce a symbolic matrix $[B, L, K]$, where $K = 8$ (vocabulary) and $L = 8$ (sequence length).
- **Symbol Decoder:** One-hot tokens are projected to dimension D via a shared linear layer. Reconstruction is performed by a two-layer MLP [`Linear`($L \cdot D$, 256) $\rightarrow$ ReLU $\rightarrow$ `Linear`(256, $C \cdot H \cdot W$)] followed by a Sigmoid activation.

Symbolic Convolutional Autoencoder A fully convolutional architecture with residual blocks used for image-to-image pre-training (50,000 examples) before symbolic bottleneck integration.

- **Backbone:** Three encoding/decoding blocks with Residual units and bilinear upsampling.
- **Bottleneck Replacement:** Post pre-training, the convolutional bottleneck is replaced by a 2-layer MLP encoder and 2-layer MLP decoder, interfaced via a Gumbel-softmax symbolic layer.

Symbolic Transformer This model replaces the standard bottleneck in the UNet backbone with a query-based discrete message-passing stage.

- **Transformer Decoder (Image-to-Symbol):**
  - **Latent Projection:** Flattened visual features are projected to $D = 512$.
  - **Query Mechanism:** $L = 8$ learnable query embeddings + positional embeddings are decoded against the visual memory using a 2-layer `TransformerDecoder` ($n\_head = 4$, $dropout = 0.2$).

  - **Output Heads:** L position-specific heads [`Dropout` $\rightarrow$ `Linear`] generate logits for vocabulary $K = 8$. Gaussian noise ($\sigma = 0.1$) is added to logits during training for regularization.
- **Transformer Encoder (Symbol-to-Image):**
  - **Sequence Processing:** Embedded symbols ($D = 128$) are processed through a 2-layer `TransformerEncoder`.
  - **Hybrid Fusion:** A composite representation is formed by concatenating the sequence mean $[B, D]$ with the flattened sequence $[B, L \cdot D]$.
  - **Bottleneck Reconstruction:** A final `Linear` layer projects this combined vector back to the original bottleneck shape (e.g., $32 \times 5 \times 5$).

Similarity Model Architecture Used for the 4-way multiple-choice evaluation by computing cosine similarity in a shared latent space ($D = 128$).

- **Image Encoder:** `Conv2d` $\rightarrow$ ReLU $\rightarrow$ `MaxPool` $\rightarrow$ `AdaptiveAvgPool`((4, 4)) $\rightarrow$ `Linear`.
- **Similarity Head:** L2-normalized feature vectors compute $s_i = \cos(f_{\text{target}}, f_{q_i})$ for $i \in \{1 \dots 4\}$.

Hyperparameters

- **Optimization:** ADAM optimizer; Learning rate 1×10^{-3} (pre-training) and 1×10^{-4} (symbolic training).
- **Regime:** 100 epochs; Early stopping (patience 10) based on 500-example validation set.
- **Transformer Config:** $D = 512$, $n\_heads = 4$, $n\_layers = 2$, $dropout = 0.2$.
- **Loss Functions:** MSE for reconstruction; Cross-Entropy for 4-way similarity classification.

D Additional Methods and LLM usage

Models were all trained on AWS G4DN-2XL

vCPUs: 8, Memory: 32 GiB, GPU: NVIDIA T4

Model weights are available at:

https://huggingface.co/datasets/CooperCognitive/mathtakestwo/tree/main/checkpoints

An LLM was used to assist in writing the code package, at the level of code lines and blocks vs. whole scripts and files. LLMs were used for grammar and wording suggestions only; all technical content and experimental design were authored by the researchers.