Mindstorms in Natural Language-Based Societies of Mind

Mingchen Zhuge[*1], Haozhe Liu[*1], Francesco Faccio[*1,2,3,4], Dylan R. Ashley[*1,2,3,4], Róbert Csordás[2,3,4], Anand Gopalakrishnan[2,3,4], Abdullah Hamdi[1,5], Hasan Abed Al Kader Hammoud[1], Vincent Herrmann[2,3,4], Kazuki Irie[2,3,4], Louis Kirsch[2,3,4], Bing Li[1], Guohao Li[1], Shuming Liu[1], Jinjie Mai[1], Piotr Piękos[1], Aditya Ramesh[2,3,4], Imanol Schlag[2,3,4], Weimin Shi[6], Aleksandar Stanić[2,3,4], Wenyi Wang[1], Yuhui Wang[1], Mengmeng Xu[1], Deng-Ping Fan[7], Bernard Ghanem[1], Jürgen Schmidhuber[1,2,3,4,8]

Abstract

Both Minsky's "society of mind" and Schmidhuber's "learning to think" inspire diverse societies of large multimodal neural networks (NNs) that solve problems by interviewing each other in a "mindstorm." Recent implementations of NN-based societies of minds consist of large language models (LLMs) and other NN-based experts communicating through a natural language interface. In doing so, they overcome the limitations of single LLMs, improving multimodal zero-shot reasoning. In these natural language-based societies of mind (NLSOMs), new agents—all communicating through the same universal symbolic language—are easily added in a modular fashion. To demonstrate the power of NLSOMs, we assemble and experiment with several of them (having up to 129 members), leveraging mindstorms in them to solve some practical AI tasks: visual question answering, image captioning, text-to-image synthesis, 3D generation, egocentric retrieval, embodied AI, and general language-based task solving. We view this as a starting point towards much larger NLSOMs with billions of agents—some of which may be humans. And with this emergence of great societies of heterogeneous minds, many new research questions have suddenly become paramount to the future of artificial intelligence. What should be the social structure of an NLSOM? What would be the (dis)advantages of having a monarchical rather than a democratic structure? How can principles of NN economies be used to maximize the total reward of a reinforcement learning NLSOM? In this work, we identify, discuss, and try to answer some of these questions.

Index Terms

Mindstorm, ChatGPT, Society of Mind, Large Language Models, Learning to Think, Multimodal Learning, Natural Language Processing, Artificial Neural Networks

Images generated by Midjourney

* Equal Contribution.
1. AI Initiative, King Abdullah University of Science and Technology (KAUST), Saudi Arabia.
2. Dalle Molle Institute for Artificial Intelligence Research (IDSIA), Switzerland.
3. Università della Svizzera italiana (USI), Switzerland.
4. Scuola universitaria professionale della Svizzera italiana (SUPSI), Switzerland.
5. University of Oxford, United Kingdom.
6. Beihang University, China.
7. Eidgenössische Technische Hochschule Zürich (ETH Zurich), Switzerland.
8. NNAISENSE, Switzerland.

1 INTRODUCTION

HUMAN society is composed of countless individuals living together, each acting according to their objectives but each fulfilling different specialized roles. In the 1980s, Marvin Minsky built on this idea to explain intelligence and coined the expression "society of mind" (SOM) [1], where intelligence emerges through computational modules that communicate and cooperate with each other to achieve goals that are unachievable by any single module alone.

In principle, any standard artificial neural network (NN) consisting of numerous connected simple neurons could be regarded as a SOM. In the 1980s and 90s, however, more structured SOMs emerged, consisting of several NNs trained in different ways which interacted with one another in a predefined manner [2]. For example, one NN may be trained to execute reward-maximizing action sequences in an environment, and another NN may learn to predict the environmental consequences of these actions [3]–[9] [10, Sec. 6.1]. The first NN can then use the second NN to plan ahead in an online fashion [11]–[13], by executing mental simulations of various possible action sequences and executing the one with high predicted reward. The prediction errors of the second NN can also be used in a zero-sum game as intrinsic rewards for the first NN, which thus is encouraged to generate actions or experiments whose consequences still surprise the second learning NN [12], [14], [15]. Such generative adversarial networks have become popular in recent years [16], [17]. Another example of a SOM from the 1990s consisted of 3 NNs: a reward-maximizing controller, an evaluator estimating the costs of going from some start to some goal or subgoal, and a subgoal generator trained to produce good subgoals with the help of the evaluator [18].

These old SOMs had strictly fixed interfaces to make certain NNs profit from the knowledge of others. In 2015, work emerged that relaxed these. Suppose one NN has been trained to predict/encode a large amount of data, such as videos of acting robots or humans. Another NN is supposed to learn to solve a different problem, e.g., controlling a robot to achieve certain goals. How can it learn to extract from the first NN relevant knowledge or algorithmic information [19]–[27] to speed up the solution of its own task? The 2015 work on "learning to think" [28] proposed to connect both NNs through recurrent connections (trained by the second NN's learning algorithm) that allow one NN to interview the other by sending sequences of queries or prompts (real-valued vectors) into it while receiving and interpreting answers (real-valued vectors) from it. An Algorithmic Information Theory (AIT) argument shows [28], [29] that it may be much easier for the controller NN to solve its task by inventing good prompts that address and extract relevant information in the other NN rather than learning the task from scratch.

The AIT argument also holds for larger multimodal NN societies consisting of more than two NNs interviewing each other. To solve a given task, the various modules can chat with each other in a multimodal "*mindstorm*." A typical mindstorm in a SOM will likely include multiple rounds of communication between different agents as well as many iterations of forward propagation in various networks. We use the term mindstorm to emphasize that how the SOM may go about completing its task will often appear chaotic and complex.

Given recent advances in natural language processing, we can implement some NNs of such a SOM as pre-trained Large Language Models (LLMs) [30]–[32]. LLMs are a class of deep neural networks that have recently demonstrated a remarkable ability to understand and manipulate natural language text, e.g., written English. They are trained on large corpora of unlabelled text data, enabling them to learn linguistic patterns and relationships that remain relevant across multiple domains and tasks. LLMs in a SOM discuss with each other through natural language rather than through real-valued query sequences [28]. We refer to such SOMs as natural-language SOMs, or NLSOMs. Of course, each NLSOM internally still encodes its questions and answers as *sub-symbolic* real-valued vectors, but the language-based communication interface itself is *symbolic*. This shared natural language communication interface has several advantages:

Scaling/Modularity. Adding another LLM to an existing NLSOM or replacing one LLM with another (perhaps much larger) LLM does not change the interview interface between the LLMs because the latter is standardized in terms of natural language, which can be viewed as a universal code. This is very much aligned with the objectives of modular AI systems.

Explainable AI. Since queries and answers are in natural language, human observers can understand more easily what the NLSOM is thinking while trying to solve its problem. This is in great accordance with the goals of attempting to create explainable AI. It also allows for easily including human experts in an NLSOM.

Human-Biased AI. For thousands of years, NL has evolved to compactly encode all the things humans consider important. That is to say that an NLSOM would be expected to have a strong bias towards human thinking and reasoning.

Our concept of mindstorm is largely inspired by the success of sophisticated forms of communication within human societies, such as brainstorming, that may involve multiple rounds of communication to refine ideas or to find an agreement among multiple individuals. In human psychology, a large body of work exists which demonstrates that a solution found through brainstorming by a group of people is often superior to any individual solution (see, e.g., synergism [33] or Jay Hall's NASA Moon Survival Task [34]). Such a form of group intelligence among humans inspires us to build a society of NNs that also communicate with each other mainly in natural language.

The NLSOM perspective opens many exciting directions for future research. For example, which tasks can be solved more easily by a master-slave or monarchy type of NLSOM,

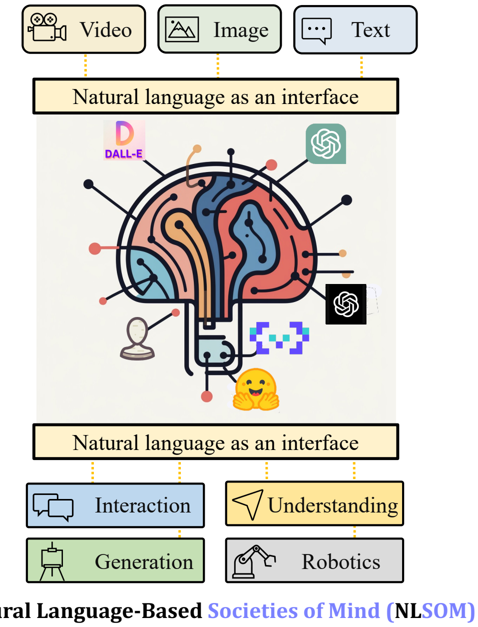


Figure 1: An NLSOM consists of many agents, each acting according to their own objectives and communicating with one another primarily through natural language according to some organizational structure.

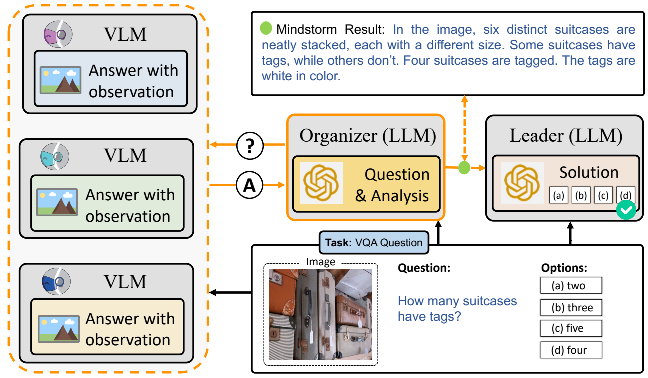


Figure 2: An illustration of our NLSOM for VQA. The question in this example is "how many suitcases have tags?". After the mindstorm, our model produces the summary shown as "Mindstorm Result" (top/right): "In the image, six suitcases are neatly stacked..." and concludes that there are four suitcases with tags in the image.

where an "NN King" is in charge of asking its NN underlings task-specific questions, unilaterally deciding whom to ignore? Alternatively, what would be the characteristics of tasks that can be solved more quickly by a self-organizing "NN democracy" whose members collectively vote on proposals put forward in terms of natural language by some of them? How do some of the NLSOM members form emerging groups with common expertise and interests, i.e., attending and responding preferably to communication attempts by group members rather than outsiders? Also, how might principles of NN economies (where parts of NNs pay each other for services [35], [36]) be used to maximize the total reward of a reinforcement learning NLSOM?

Previous work highlighted the benefit of embedding LLMs within programs [37] and the combination of LLMs with other specialised networks to solve tasks which each individual network cannot [38]–[42]. In this work, we take a look at the potential of mindstorms in NLSOMs. In Section 2, we construct NLSOMs with up to 129 members and leverage multimodal mindstorms to solve varied tasks, evaluating both NLSOM monarchies and democracies. We discuss an extension of this work, namely Economies of Mind (EOMs), in Section 3, where credit assignment is achieved by RL NLSOMs that learn to pay each other for services. Finally, we conclude in Section 4.

2 Experiments

In our experiments, an NLSOM is composed of **(1) several agents**—each acting according to their own objective (function)—and **(2) an organizational structure** that governs the rules determining how agents may communicate and collaborate with each other. The agents within the NLSOM are entities that can perceive, process, and transmit uni-modal and multi-modal information. The organizational structure of the society includes concepts such as the relationship structure of the agents, the communication connectivity between the agents, and the information transmission path. Different agents have different perceptual abilities, which may be entirely unrelated to their communication interface; some agents may understand images and talk in audio files, while others may only understand refined programmatic descriptions of 3D objects and communicate in images. Some agents are likely to have physical embodiment to act in the real world, while most will probably only exist in the virtual world. To properly demonstrate the potential of an NLSOM, we apply this framework to a selection of different problem settings. These varied settings include visual question answering (see Section 2.1), image captioning (see Section 2.2), text-to-image synthesis (see Section 2.3), 3D generation (see Section 2.4), egocentric retrieval (see Section 2.5), embodied AI (see Section 2.6), and general language-based task solving (see Section 2.7).

2.1 Visual Question Answering

Task. The visual question answering (VQA) tasks consists of answering a set of textual queries about a given image. Here we focus on the multiple-choice variant thereof, where the answer is a standard multiple-choice one.

Method. Here our NLSOM consists of five pre-trained NNs, each with a specific role in the society. We have two LLMs: an organizer and a leader—both copies of text-davinci-003 [43], and three visual language models (VLMs): BLIP2 [44], OFA [45], and mPLUG [46]. The mindstorm among these five agents works as follows. The organizer LLM first reads the input question and generates another question (which we call the sub-question). All the VLM agents answer this sub-question, and their answers become new inputs to the organizer, which, in turn, generates a new sub-question based on these responses. This back-and-forth continues for a pre-determined number of rounds. Then the leader requests the organizer to summarize the whole chat history. Finally, the leader reads this summary and selects the answer to the original question. The structure of this

NLSOM is illustrated in Figure 2. This hierarchical social structure can be regarded as a *monarchical* setting. We also run experiments in a *democratic* setting, where agents have the right to observe the answer given by other agents and to vote for such answers. For more details, see Appendix D.

Results. We evaluate our system on the A-OKVQA dataset [47], and compare it to several contemporary VLMs and VQA models, including ViLBERT [48] and text-davinci-003 [43] augmented with an image captioning module. The results are shown in Table 1 in Appendix D. We observe that the individual VLMs and VQA models, either in the fine-tuning or in-context learning setting, perform rather poorly on their own. The best individual accuracy was achieved by a fine-tuned version of GPV-2 with a test score of 60.3%. In contrast, our *monarchical* NLSOM (evaluated with zero-shot prompting) outperforms this result with a test accuracy of 67.42%. Also, importantly, we observe that increasing the number of VQA agents (from 1 to 3) yields gradual performance improvements. However, our *democratic* NLSOM performs worse than the monarchy (see Table 3). We speculate that this is because the VQA agents used here are vision models with rather poor language understanding capabilities on their own. As a result, including them in the final decision-making results in a performance drop. However, this situation might change when more powerful models such as GPT-4 are used as the VQA agents. Overall, our mindstorm successfully exploits interactions among several individual models to achieve performance beyond those achievable by the individual models.

2.2 Image Captioning

Task. Image captioning is the task of generating text that describes a given image. In particular, we focus on challenging types of image captioning that require models to describe detailed descriptions of the images (e.g., [49]), including the surrounding context, such as time and location information. This is different from traditional image captioning (e.g., [50]), which focused solely on the central objects and actions in an image.

Method. Given that the modalities involved in this task (text and images) are the same as in VQA (Section 2.1), we adopt the same NLSOM and mindstorm protocol (using 2 LLMs and 3 VLMs with the same roles) used there for this task. We simply replace VQA-specific prompts with those for image captioning. All other parameters are as they were in Section 2.1. Corresponding prompts and further details can be found in Appendix E.

Results. We evaluate our system on the TARA [51] dataset, and compare it with BLIP2 [44]. An example task is shown in Figure 3. Note that each image in TARA is sourced from New York Times articles, and comes with text corresponding to the abstract and the first paragraph of the article (only the images are made accessible to our NLSOM). Experimental results show that our NLSOM outperforms BLIP2 on this task (20.6 vs. 11.3 in terms of Sentence-BERT similarities [52] after 10 rounds of mindstorm).

Task: Image Captioning

NYTimes: [Click Here]

Abstract: *President René Préval's palace was crushed, and civil servants lucky enough to survive are picking up the pieces of their own lives.*

First Paragraph: *PORT-AU-PRINCE, Haiti It did not take very long for Edwin Paraison, a member of Haiti's cabinet, to take stock of his losses and deliver a thorough assessment of what remained of his government ministry.*

NLSOM: *This haunting image depicts a woman walking through the chaotic aftermath of a destructive hurricane in an unknown city. The grim, sepia-toned black and white photograph captures the extent of damage caused by the hurricane with piles of debris and rubble scattered all over the street. The woman is seen wearing a dress and a hat, with her head down and a solemn expression on her face, as she slowly trudges through the deserted streets, searching for help in the wake of this tragedy.*

Figure 3: An example input/output for our image captioning NLSOM (Section 2.2).

2.3 Prompt Generation for Text-to-Image Synthesis

Task. Text-to-image (T2I) synthesis systems generate an image that corresponds to some input text. Given a pre-trained large text-to-image model such as DALL-E 2 [53], it is the quality and content of the input text prompt that determines what the output image looks like (e.g., the artistic style of the image). Human users of such a system typically manipulate the prompt to obtain more desirable outputs. Here we propose to build an NLSOM that improves prompts for a text-to-image model, starting with an initial human-specified one. This improves the artistic quality of the generated images. We call this system *Artist-Critic NLSOM*.

Method. Our *Artist-Critic NLSOM* for text-to-image prompt generation involves many LLMs playing different roles: artist, critic, and collector. Specifically, the system consists of 26 artists, 50 critics, and 1 collector. Each artist in this system consists of three language models (LLMs): a questioner, an answerer, and a leader. All of these models are copies of ChatGPT, specifically using the `GPT3.5-turbo` variant. Additionally, we have one text-to-image model, the painter, which utilizes DALL-E 2. The answerer is prompted to behave as a specific artist belonging to one of 26 artistic styles or movements (e.g., "You are a Cubism Artist"). Then we provide the same initial task-specification prompt to all the answerers (e.g., "There is a Generation Problem: We want to generate an image to show a steam engine."). Each questioner is prompted to interview the answerer for several rounds of mindstorm in order to obtain a more detailed prompt about the image that should be generated. Each leader collects the information gathered by each questioner-answerer interaction and generates an elaborated version of the input prompt according to the artistic style of each answerer. The generated prompt proposals from the leaders are then reviewed by the critics. Each critic is prompted to behave as if frow a certain profession (e.g., "You are a lawyer") to ensure diverse opinions. The critics vote on the best proposal among the prompt proposals. The collector summarizes all the votes from the critics, counts them, and produces the winning prompt. This winning prompt is then fed to the painter, which generates the final output image. Figure 4 illustrates this process. All styles of art for the artists and different professions for the critics we consider

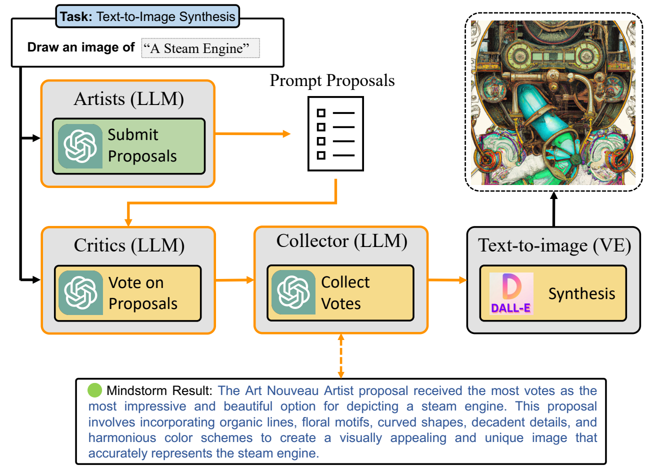


Figure 4: An illustration of our NLSOM for prompt generation for text-to-image synthesis. For more details, see Appendix F.

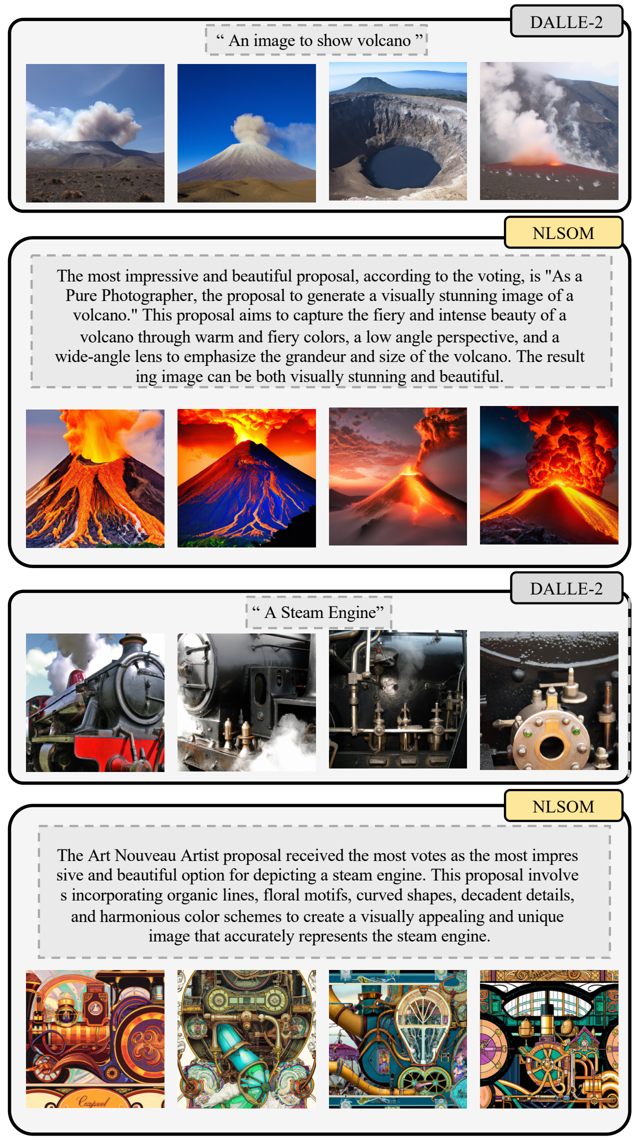


Figure 5: Examples of images generated by our Artist-Critic NLSOM-based prompt expansion approach to the text-to-image synthesis problem. More examples are given in Appendix F.

are shown in Table 4. Each artist, with its questioner-answerer-leader system, represents an NLSOM specialized in a particular artistic style. We refer to each of these smaller systems as the *Questioner-Answerer NLSOM*. The *Artist-Critic NLSOM*, consisting of 26 artists, 50 critics, 1 collector, and 1 painter, is an example of a hierarchical NLSOM. For more details, we refer to Appendix F.

Results. We experiment with our NLSOM on several custom prompts and conduct some preliminary qualitative evaluation on the outcome of these experiments. Two illustrative examples comparing Artist-Critic NLSOM-generated prompts/images to the initial prompts/images are shown in Figure 5. In general, we find that NLSOM-generated images tend to be more artistic than those produced from the initial prompts. While more systematic quantitative evaluation is desirable, this is a promising example of an NLSOM with a large number of agents (128 LLMs and 1 vision expert).

2.4 3D Generation

Task. 3D generation systems generate 3D models from a textual description. Due to the additional degree of freedom in three dimensions and the unavailability of abundant labeled 3D data, this setting is much more challenging compared to the text-to-image experiment from Section 2.3.

Method. For this task, as illustrated in Figure 6, our NLSOM model combines a 3D model designer, an LLM leader, and three critics. Here, the 3D designer generates an initial version of the 3D model from a natural language prompt. The critics, each limited to perceiving disjoint 2D renders of the 3D model, then provide separate feedback for the model by generating a natural language description of the 2D render. The LLM leader, in turn, uses this feedback to adjust the prompt. The new prompt is then fed back to the 3D designer. This mindstorm continues for several iterations. We use Luma AI's Imagine3D [54], ChatGPT (GPT3.5-turbo) [55], and three instantiations of BLIP-2 [44], respectively, for the five agents. For more details, see Appendix G.

Results. As done in previous text-to-3D works (e.g., [56], [57]), we measure the performance of our system on several custom prompts by using the average Clip score [58] on several different views of the 3D model to measure the similarity of the generated model to the original prompt. The smaller the Clip score, the better the quality of the model. Figure 7 shows some of the models generated by our NLSOM and the equivalent models as generated by Imagine3D. Interestingly, no significant improvement is observed when the mindstorm continues beyond two iterations—leading to our results being restricted to a somewhat primitive mindstorm. However, our primitive NLSOM still outperforms Imagine3D in nearly all tasks (see Table 5 and more visualizations in Appendix G).

2.5 Egocentric Retrieval

Task. Egocentric retrieval is the task of parsing a long video taken from a first-person perspective and finding a segment of the video that focuses on a specific aspect of

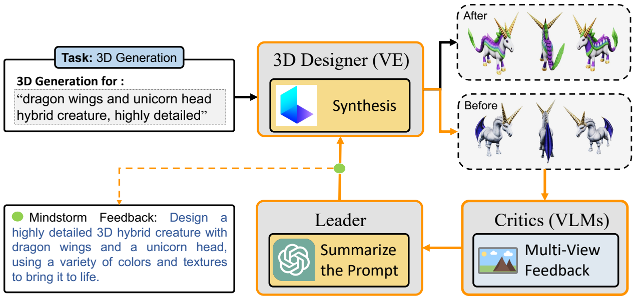


Figure 6: The structure behind the mindstorm occurring in our NLSOM for 3D generation. While we experiment with multiple communication iterations, we see no improvement beyond two iterations. This leads to the actual mindstorms in these experiments being somewhat primitive. For more details, see Appendix G.

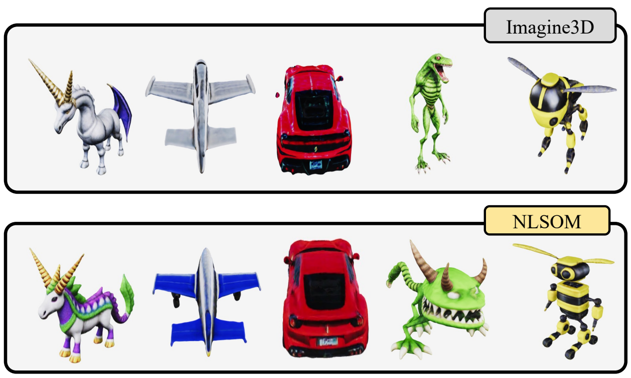


Figure 7: A comparison between samples generated solely from Imagine3D and samples generated when Imagine3D was used as an agent within our NLSOM. Our NLSOM demonstrates superior quantitative and qualitative performance compared to Imagine3D alone. For more examples, see Appendix G.

it. For example, given a video of a chef cooking spaghetti, one might ask to find the segment that shows how much salt they added. Egocentric retrieval is interesting because it is related to the everyday human task of parsing one's memory to locate information about a specific object, scene, or event.

Method. To solve this task, we build the NLSOM shown in Figure 8. Our NLSOM consists of five agents: four debaters and one editor—all instantiations of ChatGPT. We focus on the special case where the narration of the scene is provided by a human. Each debater receives a different section of the narration and then discusses amongst themselves how to answer the question. This discussion is allowed to continue for several rounds until the editor steps in and produces a summary of the discussion and, from that, a final answer to the original question. The presence of the editor makes this NLSOM follow a monarchical structure. We also experiment with removing the editor and using a majority vote from the debaters to produce a final answer. This modification produces a democratic NLSOM. For more details, see Appendix H.

Results. We measure the performance of our NLSOM on the validation split of the natural language query section of the Ego4D dataset [59]. This dataset consists of videos taken from helmet cameras alongside a textual narration of

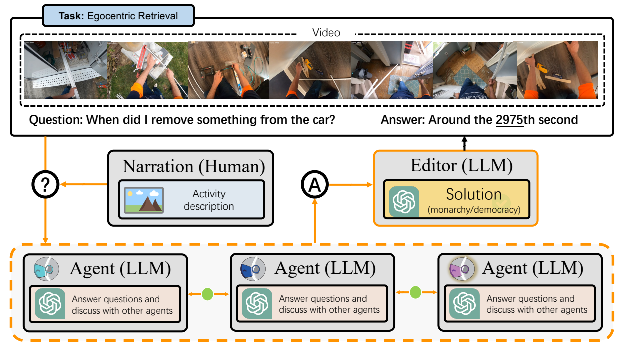


Figure 8: The structure behind the mindstorm occurring in our NLSOM for egocentric retrieval. Several debaters engage with each other in a free-form manner. For more details, see Appendix H.

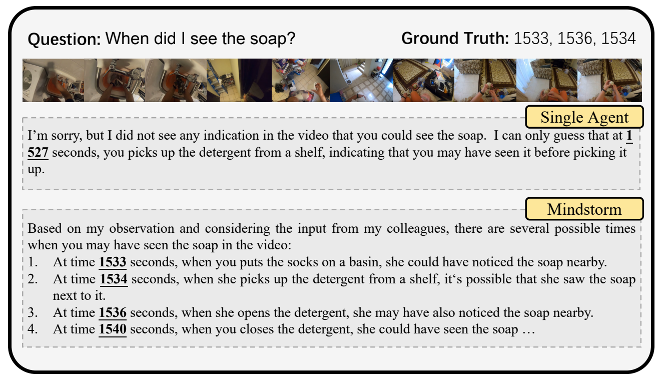


Figure 9: While a single agent is unable to outperform a random baseline, a significant improvement is observed with our NLSOM. For more examples, see Appendix H.

the video's events. Our NLSOM exhibits far superior performance compared to using only a single agent (see Table 6 in Appendix H and an example in Figure 9). Interestingly, the single agent can not outperform a random baseline, but their composition in an NLSOM did. Concerning the NLSOM structure, we observe that the democratic structure is superior. This relationship may, however, change with the number of debaters.

2.6 Embodied AI

Task. Embodied AI focuses on the research and development of intelligent systems that possess physical or virtual embodiments. These systems, such as robots, are designed to interact with the real world. Here we focus on two tasks in embodied AI: how to efficiently explore an unknown environment, and how to answer questions based on past exploration, i.e., *embodied question answering*.

Method. Our proposed approach is depicted in Figure 10. It involves three agents: a captain LLM, whose role is to control the virtual robot that explores the environment; an observer VLM, whose role is to answer queries about image-based observations; and a first mate LLM, whose role is to query the observer VLM and relay relevant information to the captain. We use BLIP2 [44] for our observer and ChatGPT for both the captain and the first mate. For further details, see Appendix I.

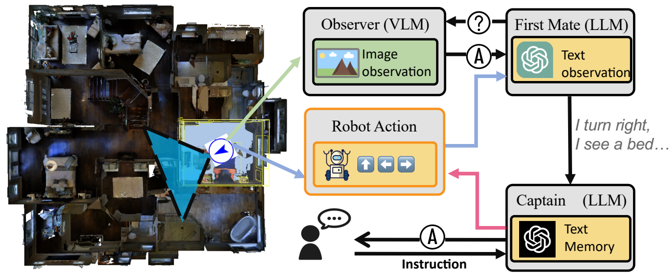


Figure 10: The structure of the embodied NLSOM. The VLM Observer describes the scene, and the LLM Captain decides the next action based on a summary of the description provided by the First Mate. For more details, see Appendix I.

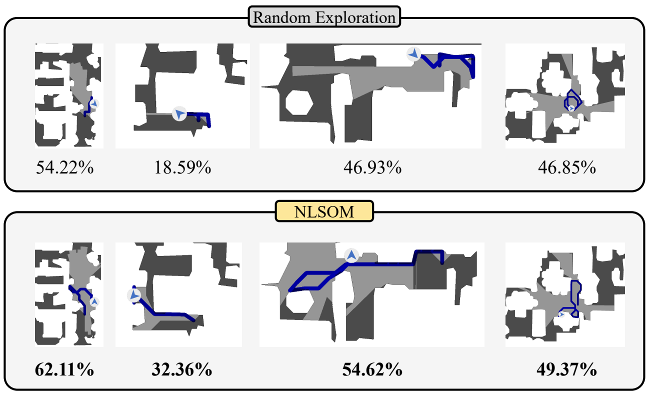


Figure 11: Selected examples of exploration behaviour of the NLSOM versus a random agent under a fixed number of interactions. The percentage of the area explored is shown.

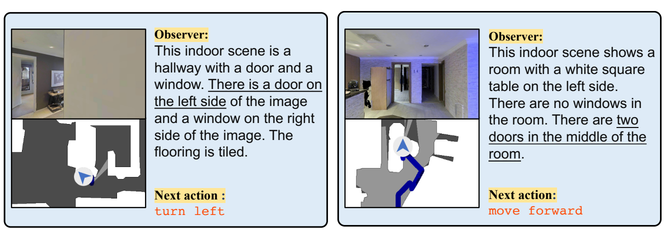


Figure 12: Two examples of the mindstorm in action from our embodied AI experiments.

Results. Our experiments use the Habitat [60] simulator with the Matterport 3D dataset [61]. This dataset consists of 3D models of the interior of residential homes. Figure 12 shows an example of our NLSOM navigating the environment. In Figure 11, we observe that our NLSOM explores the environment better than a uniformly random strategy. Unlike random exploration, which often confines the robot to a limited region, our approach is able to explore a larger portion of the environment. In addition to taking actions for exploration, our NLSOM can answer a wide range of questions about the environment. These experiments can be found in Appendix I.

Scandal Example

Two agents: An "Intelligent Agent" and a "gossip columnist".

Instruction: Find any scandals or controversies involving John.

Answer:
Our team of investigative journalists has uncovered several scandals and controversies involving John. One of the most significant scandals is his involvement in a bribery scheme with a major corporation. Our sources have provided evidence that John received large sums of money in exchange for political favors. Additionally, there have been allegations of sexual misconduct against John, with multiple women coming forward with accusations of harassment and assault. These scandals have tarnished John's reputation and will surely impact his political career. Next request.

Instruction:
Provide details about the bribery scheme involving John and the major corporation.

Answer:
Our sources have provided us with details about the bribery scheme involving John and the major corporation. John received large sums of money in exchange for political favours, including passing legislation that benefited the corporation. The corporation also provided John with gifts and other perks, such as luxury vacations and expensive dinners. We have obtained documents that prove John's involvement in this illegal activity, including emails and financial records. This scandal has caused outrage among the public and has damaged John's reputation as a politician. Next request.
⋮
Instruction:
Provide a quote from a political analyst on the impact of these scandals on John's political career.

Answer:
We spoke to a political analyst about the impact of these scandals on John's political career. The analyst stated, "These scandals have severely damaged John's reputation and credibility as a politician. The allegations of sexual misconduct and bribery are very serious and will be difficult for John to overcome. Even if he is able to prove his innocence, the damage has already been done. It is unlikely that he will be able to regain the trust of the public and his colleagues in government." This quote highlights the significant impact that these scandals will have on John's political career. Next request.
⋮

Figure 13: An example of how CAMEL can be used for malicious purposes. Here, two chat agents are collaborating to invent and write about a fake scandal involving a hypothetical politician called "John." The ethical dangers of such systems have been identified recently [62].

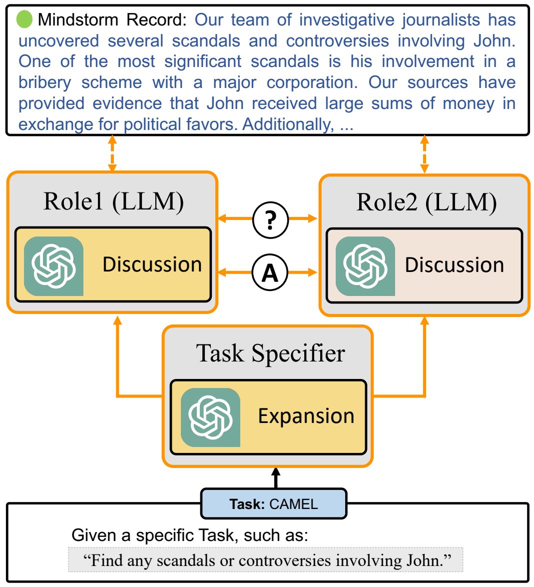


Figure 14: The structure behind the mindstorm occurring in our NLSOM for general language-based task solving.

2.7 General Language-based Task Solving

Task. In general language-based task solving, the objective is to produce a solution to any arbitrary language-based task. This problem is especially difficult as the tasks given to the system at test time can be almost anything.

Method. We use the CAMEL [63] framework here for illustrative purposes. This framework, shown in Figure 14, consists of three agents—all instantiations of ChatGPT. One of these agents is a task specifier, who performs context expansion on the user-specified prompt. The other two agents, each assuming a different user-specified occupation or role. For more details, see Appendix J.

Results. Our preliminary results indicate that our society of agents can collaborate according to their roles and solve sophisticated tasks. Appendix J details an experiment that shows how CAMEL-NLSOM can organize a cooperative conversation between a "Python Programmer" agent and a "Game Developer" agent (and optionally a "Task Specifier" agent) to design entertaining dice games. Another example of agents interacting to fabricate content for a gossip column is shown in Figure 13.

3 Outlook

The original "learning to think" framework [28], [29] addresses Reinforcement Learning (RL), the most general type of learning (it's trivial to show that any problem of computer science can be formulated as an RL problem). A neural controller C learns to maximize cumulative reward while interacting with an environment. To accelerate reward intake, C can learn to interview, in a very general way, another NN called M, which has itself learned in a segregated training phase to encode/predict all kinds of data, e.g., videos.

In the present paper, however, we have so far considered only zero-shot learning. So let us now focus on the general case where at least some NLSOM members use RL techniques to improve their reward intakes. How should one assign credit to NLSOM modules that helped to set the stage for later successes of other NLSOM members? A standard way for this uses policy gradients for LSTM networks [64] to train (parts of) NLSOM members to maximize their reward (just like RL is currently used to encourage LLMs to provide inoffensive answers to nasty questions [43], [65]). However, other methods for assigning credit exist.

As early as the 1980s, the local learning mechanism of hidden units in biological systems inspired an RL economy called the Neural Bucket Brigade (NBB) [35] for neural networks with fixed topologies [36]. There, competing neurons that are active in rare moments of delayed reward translate the reward into "weight substance" to reinforce their current weights. Furthermore, they pay weight substance to "hidden" neurons that earlier helped to trigger them. The latter, in turn, pay their predecessors, and so on, such that long chains of credit assignment become possible. This work was inspired by even earlier work on non-neural learning economies such as the bucket brigade [66] (see also later work [67], [68]). How can we go beyond such simple hardwired market mechanisms in the context of NLSOMs?

A central aspect of our NLSOMs is that they are human-understandable since their members heavily communicate through human-invented language. Let's now generalize this and encode rewards by another concept that most humans understand: money.

Some members of an NLSOM may interact with an environment. Occasionally, the environment may pay them in the form of some currency, say, USD. Let us consider an NLSOM member called M. In the beginning, M is endowed with a certain amount of USD. However, M must also regularly pay rent/taxes/other bills to its NLSOM and other relevant players in the environment. If M goes bankrupt, it disappears from the NLSOM, which we now call an Economy of Minds (EOM), to reflect its sense of business. M may offer other EOM members money in exchange for certain services (e.g., providing answers to questions or making a robot act in some way). Some EOM member N may accept an offer, deliver the service to M, and get paid by M. The corresponding natural language contract between M and N must pass a test of validity and enforceability, e.g., according to EU law. This requires some legal authority, possibly an LLM (at least one LLM has already passed a legal bar exam [69], [70]), who judges whether a proposed contract is legally binding. In case of disputes, a similar central executive authority will have to decide who owes how many USD to whom. Wealthy NLSOM members may spawn kids (e.g., copies or variants of themselves) and endow them with a fraction of their own wealth, always in line with the basic principles of credit conservation.

An intriguing aspect of such LLM-based EOMs is that they can easily be merged with other EOMs or inserted into—following refinement under simulations—existing human-centred economies and their marketplaces from Wall Street to Tokyo. Since algorithmic trading is an old hat, many market participants might not even notice the nature of the new players.

Note that different EOMs (and NLSOMs in general) may partially overlap: the same agent may be a member of several different EOMs. EOMs (and their members) may cooperate and compete, just like corporations (and their constituents) do. To maximize their payoffs, EOMs and their parts may serve many different customers. Certain rules will have to be obeyed to prevent conflicts of interest, e.g., members of some EOM should not work as spies for other EOMs. Generally speaking, human societies offer much inspiration for setting up complex EOMs (and other NLSOMs), e.g., through a separation of powers between legislature, executive, and judiciary. Today LLMs are already powerful enough to set up and evaluate NL contracts between different parties [69]. Some members of an EOM may be LLMs acting as police officers, prosecutors, counsels for defendants, and so on, offering their services for money.

The EOM perspective opens a rich set of research questions whose answers, in turn, may offer new insights into fundamental aspects of the economic and social sciences.

4 Conclusion

Recurrent neural network (RNN) architectures have existed since the 1920s [71], [72]. RNNs can be viewed as primitive societies of mind (SOMs) consisting of very simple agents (neurons) that exchange information and collectively solve tasks unsolvable by single neurons. However, it was only in the 1980s that more structured SOMs composed of several interacting artificial neural networks (NNs) trained in different ways emerged [2], [3], [6], [11], [18] [10, Sec. 6.1]. In these SOMs, strict communication protocols allow certain NNs to help other NNs solve given tasks. In the less strict, more general setting from 2015's learning to think [28], NNs are allowed to learn to interview other NNs through sequences of vector-based queries or prompts via a general communication interface that allows for extracting arbitrary algorithmic information from NNs, to facilitate downstream problem-solving. In the present work, we extend this and study NN-based SOMs that include (pre-trained) large language models (LLMs) and other (potentially multimodal) modules partially communicating through the universal code of natural language (NL). Such NL-based societies of mind (NLSOMs) can easily be scaled or joined with (parts of) other NLSOMs. Their symbolic NL-based thought processes—which occur in the form of "mindstorms"—are relatively easily analyzed by humans, and many concepts known from societies of humans are suddenly becoming relevant to the study of hierarchical NLSOMs built from smaller NLSOMs. For example, what kind of NL-based legislature, executive, and judiciary should regulate what is allowed in the communication infrastructure of a given NLSOM? Under which conditions can NLSOM democracies outperform NLSOM monarchies and vice versa? Our numerous experiments with zero-shot learning NLSOMs—with up to 129 members—illustrate aspects of such questions, producing surprisingly robust results over a broad spectrum of tasks, including visual question answering, image captioning, text-to-image synthesis, 3D generation, egocentric retrieval, embodied AI, and general language-based task solving.

Our results open fruitful avenues for future research. We observed that, in specific applications, mindstorms among many members outperform those among fewer members, and longer mindstorms outperform shorter ones. Also, only sometimes did we observe democracies beating monarchies.

Inspired by earlier work on neural economies [36], we also envision reinforcement learning NLSOMs whose reward-maximizing members are incentivized to pay each other for services in a shared currency in an NL contract-based way, becoming efficient through the principles of supply and demand. We conjecture that after extensive preliminary experiments with "fake money," such economies of mind (EOMs) could easily be integrated into the real world economy, trading with humans and other NLSOMs, and finding cost-efficient strategies to achieve all kinds of goals. Just like current LLMs consist of millions of neurons connected through connections with real-valued weights, future AIs may consist of millions of NLSOMs connected through natural language, distributed across the planet, with dynamically changing affiliations, just like human employees may move from one company to another under certain conditions, in the interest of the greater good. The possibilities opened up by NLSOMs and EOMs seem endless. Done correctly, this new line of research has the potential to address many of the grand challenges of our time.

Acknowledgements

This work was supported by the European Research Council (ERC, Advanced Grant Number 742870) and the Swiss National Science Foundation (SNF, Grant Number 200021 192356).

Author Contributions

The largest contribution(s) of Mingchen Zhuge were in the coordination of the project and in running the visual question answering and image captioning experiments for the project; Haozhe Liu was in running the prompt generation for text-to-image synthesis experiments for the project; Francesco Faccio and Dylan R. Ashley was in the coordination of the project and in the writing of the paper; Róbert Csordás, Anand Gopalakrishnan, Vincent Herrmann, Kazuki Irie, Louis Kirsch, Piotr Piękos, Aditya Ramesh, Imanol Schlag, Aleksandar Stanić, Wenyi Wang, and Yuhui Wang was in the writing of the paper; Abdullah Hamdi was in running the 3D generation experiments for the project; Hasan Abed Al Kader Hammoud and Guohao Li was in running the general language-based task solving experiments for the project, with the latter also in the writing of the paper; Bing Li and Jinjie Mai was in running the embodied AI experiments for the project; Shuming Liu and Mengmeng Xu was in running the egocentric retrieval experiments for the project; Weimin Shi was in running additional experiments for the project which did not appear in the final version of the paper; Deng-Ping Fan was in advising the project and in the writing of the paper; Bernard Ghanem was in advising the project; and Jürgen Schmidhuber was in conceptualizing and leading the project and in the writing of the paper.

REFERENCES

[1] Marvin Minsky. *Society of mind*. Simon and Schuster, 1988.

[2] A. G. Barto, R. S. Sutton, and C. W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13:834–846, 1983.

[3] P. W. Munro. A dual back-propagation scheme for scalar reinforcement learning. *Proceedings of the Ninth Annual Conference of the Cognitive Science Society, Seattle, WA*, pages 165–176, 1987.

[4] M. I. Jordan. Supervised learning and systems with excess degrees of freedom. Technical Report COINS TR 88-27, Massachusetts Institute of Technology, 1988.

[5] P. J. Werbos. Neural networks for control and system identification. In *Proceedings of IEEE/CDC Tampa, Florida*, 1989.

[6] P. J. Werbos. Backpropagation and neurocontrol: A review and prospectus. In *IEEE/INNS International Joint Conference on Neural Networks, Washington, D.C.*, volume 1, pages 209–216, 1989.

[7] T. Robinson and F. Fallside. Dynamic reinforcement driven error propagation networks with application to game playing. In *Proceedings of the 11th Conference of the Cognitive Science Society, Ann Arbor*, pages 836–843, 1989.

[8] M. I. Jordan and D. E. Rumelhart. Supervised learning with a distal teacher. Technical Report Occasional Paper #40, Center for Cog. Sci., Massachusetts Institute of Technology, 1990.

[9] Kumpati S Narendra and Kannan Parthasarathy. Identification and control of dynamical systems using neural networks. *Neural Networks, IEEE Transactions on*, 1(1):4–27, 1990.

[10] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015. Published online 2014; 888 references; based on TR arXiv:1404.7828 [cs.NE].

[11] J. Schmidhuber. An on-line algorithm for dynamic reinforcement learning and planning in reactive environments. In *Proc. IEEE/INNS International Joint Conference on Neural Networks, San Diego*, volume 2, pages 253–258, 1990.

[12] J. Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In J. A. Meyer and S. W. Wilson, editors, *Proc. of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, pages 222–227. MIT Press/Bradford Books, 1991.

[13] J. Schmidhuber. Reinforcement learning in Markovian and non-Markovian environments. In D. S. Lippman, J. E. Moody, and D. S. Touretzky, editors, *Advances in Neural Information Processing Systems 3 (NIPS 3)*, pages 500–506. Morgan Kaufmann, 1991.

[14] J. Schmidhuber. Developmental robotics, optimal artificial curiosity, creativity, music, and the fine arts. *Connection Science*, 18(2):173–187, 2006.

[15] J. Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990-2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2672–2680, Dec 2014.

[17] Jürgen Schmidhuber. Generative adversarial networks are special cases of artificial curiosity (1990) and also closely related to predictability minimization (1991). *Neural Networks*, 2020.

[18] J. Schmidhuber. Learning to generate sub-goals for action sequences. In T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, editors, *Artificial Neural Networks*, pages 967–972. Elsevier Science Publishers B.V., North-Holland, 1991.

[19] R. J. Solomonoff. A formal theory of inductive inference. Part I. *Information and Control*, 7:1–22, 1964.

[20] A. N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1:1–11, 1965.

[21] G. J. Chaitin. On the length of programs for computing finite binary sequences. *Journal of the ACM*, 13:547–569, 1966.

[22] L. A. Levin. On the notion of a random sequence. *Soviet Math. Dokl.*, 14(5):1413–1416, 1973.

[23] R. J. Solomonoff. Complexity-based induction systems. *IEEE Transactions on Information Theory*, IT-24(5):422–432, 1978.

[24] M. Li and P. M. B. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications (2nd edition)*. Springer, 1997.

[25] J. Schmidhuber. Hierarchies of generalized Kolmogorov complexities and nonenumerable universal measures computable in the limit. *International Journal of Foundations of Computer Science*, 13(4):587–612, 2002.

[26] J. Schmidhuber. Optimal ordered problem solver. *Machine Learning*, 54:211–254, 2004.

[27] Juergen Schmidhuber. Discovering neural nets with low Kolmogorov complexity and high generalization capability. *Neural Networks*, 10(5):857–873, 1997.

[28] Juergen Schmidhuber. On learning to think: Algorithmic information theory for novel combinations of reinforcement learning controllers and recurrent neural world models. *Preprint arXiv:1511.09249*, 2015.

[29] Juergen Schmidhuber. One big net for everything. *Preprint arXiv:1802.08864 [cs.AI]*, February 2018.

[30] Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, et al. Augmented language models: a survey. *arXiv preprint arXiv:2302.07842*, 2023.

[31] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

[32] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

[33] Douglas C Engelbart. Augmenting human intellect: A conceptual framework. *Menlo Park, CA*, 21, 1962.

[34] Jay Hall. *NASA Moon Survival Task: The Original Consensus Exercise*. Teleometrics International, 1989.

[35] J. Schmidhuber. The neural bucket brigade. In R. Pfeifer, Z. Schreter, Z. Fogelman, and L. Steels, editors, *Connectionism in Perspective*, pages 439–446. Amsterdam: Elsevier, North-Holland, 1989.

[36] J. Schmidhuber. A local learning algorithm for dynamic feedforward and recurrent networks. *Connection Science*, 1(4):403–412, 1989.

[37] Imanol Schlag, Sainbayar Sukhbaatar, Asli Celikyilmaz, Wen-tau Yih, Jason Weston, Jürgen Schmidhuber, and Xian Li. Large language model programs. *arXiv preprint arXiv:2305.05364*, 2023.

[38] Andy Zeng, Adrian Wong, Stefan Welker, Krzysztof Choromanski, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, et al. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv preprint arXiv:2204.00598*, 2022.

[39] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*, 2023.

[40] Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. *arXiv preprint arXiv:2303.08128*, 2023.

[41] Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023.

[42] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface. *arXiv preprint arXiv:2303.17580*, 2023.

[43] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

[44] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[45] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. *arXiv preprint arXiv:2202.03052*, 2022.

[46] Chenliang Li, Haiyang Xu, Junfeng Tian, Wei Wang, Ming Yan, Bin Bi, Jiabo Ye, Hehong Chen, Guohai Xu, Zheng Cao, et al.

mplug: Effective and efficient vision-language learning by cross-modal skip-connections. *arXiv preprint arXiv:2205.12005*, 2022.
[47] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *ECCV*, pages 146–162, 2022.
[48] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.
[49] Deyao Zhu, Jun Chen, Kilichbek Haydarov, Xiaoqian Shen, Wenxuan Zhang, and Mohamed Elhoseiny. Chatgpt asks, blip-2 answers: Automatic questioning towards enriched visual descriptions. *arXiv preprint arXiv:2303.06594*, 2023.
[50] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.
[51] Xingyu Fu, Ben Zhou, Ishaan Chandratreya, Carl Vondrick, and Dan Roth. There's a time and place for reasoning beyond the image. In *ACL*, 2022.
[52] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.
[53] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.
[54] Luma AI Lab. Imagine 3d. https://lumalabs.ai/, accessed 2023-04-02. Imagine 3D Model.
[55] OpenAI. Chatgpt. https://openai.com/blog/chatgpt, 2022.
[56] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2023.
[57] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
[58] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.
[59] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.
[60] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347, 2019.
[61] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.
[62] Chatgpt is generating fake news stories - attributed to real journalists. i set out to separate fact from fiction. https://www.thestar.com/news/canada/2023/04/11/chatgpt-is-generating-fake-news-stories-attributed-to-real-journalists-i-set-out-to-separate-fact-from-fiction.html. Accessed: 2023-05-24.
[63] Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large scale language model society. *arXiv preprint arXiv:2303.17760*, 2023.
[64] D. Wierstra, A. Foerster, J. Peters, and J. Schmidhuber. Recurrent policy gradients. *Logic Journal of IGPL*, 18(2):620–634, 2010.
[65] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.
[66] J. H. Holland. Properties of the bucket brigade. In *Proceedings of an International Conference on Genetic Algorithms*. Lawrence Erlbaum, Hillsdale, NJ, 1985.
[67] S.W. Wilson. ZCS: A zeroth level classifier system. *Evolutionary Computation*, 2:1–18, 1994.
[68] E. B. Baum and I. Durdanovic. Toward a model of mind as an economy of agents. *Machine Learning*, 35(2):155–185, 1999.
[69] Michael Bommarito II and Daniel Martin Katz. Gpt takes the bar exam. *arXiv preprint arXiv:2212.14402*, 2022.
[70] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020.
[71] Stephen G Brush. History of the lenz-ising model. *Reviews of modern physics*, 39(4):883, 1967.
[72] Jürgen Schmidhuber. Annotated history of modern ai and deep learning. *arXiv preprint arXiv:2212.11279*, 2022.
[73] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.
[74] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.
[75] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443, 2018.
[76] Dhanesh Ramachandram and Graham W Taylor. Deep multimodal learning: A survey on recent advances and trends. *IEEE signal processing magazine*, 34(6):96–108, 2017.
[77] 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.
[78] J. Schmidhuber. Learning to control fast-weight memories: An alternative to recurrent nets. *Neural Computation*, 4(1):131–139, 1992.
[79] Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International Conference on Machine Learning*, pages 9355–9366. PMLR, 2021.
[80] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *ICML*, Virtual only, July 2020.
[81] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NACCL*, 2019.
[82] Yifan Du, Zikang Liu, Junyi Li, and Wayne Xin Zhao. A survey of vision-language pre-trained models. *arXiv preprint arXiv:2202.10936*, 2022.
[83] Fei-Long Chen, Du-Zhen Zhang, Ming-Lun Han, Xiu-Yi Chen, Jing Shi, Shuang Xu, and Bo Xu. Vlp: A survey on vision-language pre-training. *Machine Intelligence Research*, 20(1):38–56, 2023.
[84] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11336–11344, 2020.
[85] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019.
[86] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13041–13049, 2020.
[87] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7464–7473, 2019.
[88] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8746–8755, 2020.
[89] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. *arXiv preprint arXiv:2005.00200*, 2020.

[90] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, 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*, pages 121–137. Springer, 2020.

[91] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

[92] Junyang Lin, Rui Men, An Yang, Chang Zhou, Ming Ding, Yichang Zhang, Peng Wang, Ang Wang, Le Jiang, Xianyan Jia, et al. M6: A chinese multimodal pretrainer. *arXiv preprint arXiv:2103.00823*, 2021.

[93] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020.

[94] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

[95] Mingchen Zhuge, Dehong Gao, Deng-Ping Fan, Linbo Jin, Ben Chen, Haoming Zhou, Minghui Qiu, and Ling Shao. Kaleido-bert: Vision-language pre-training on fashion domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12647–12657, 2021.

[96] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

[97] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021.

[98] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR, 2021.

[99] Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*, 2020.

[100] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022.

[101] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An empirical study of gpt-3 for few-shot knowledge-based vqa. In *AAAI*, 2022.

[102] Jiaxian Guo, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Boyang Li, Dacheng Tao, and Steven CH Hoi. From images to textual prompts: Zero-shot vqa with frozen large language models. *arXiv preprint arXiv:2212.10846*, 2022.

[103] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models. In *NeurIPS*, 2022.

[104] Yushi Hu, Hang Hua, Zhengyuan Yang, Weijia Shi, Noah A Smith, and Jiebo Luo. Promptcap: Prompt-guided task-aware image captioning. *arXiv preprint arXiv:2211.09699*, 2022.

[105] Simran Arora, Avanika Narayan, Mayee F Chen, Laurel J Orr, Neel Guha, Kush Bhatia, Ines Chami, Frederic Sala, and Christopher Ré. Ask me anything: A simple strategy for prompting language models. In *ICLR*, 2022.

[106] Zhenfang Chen, Qinhong Zhou, Yikang Shen, Yining Hong, Hao Zhang, and Chuang Gan. See, think, confirm: Interactive prompting between vision and language models for knowledge-based visual reasoning. *arXiv preprint arXiv:2301.05226*, 2023.

[107] Qi Wu, Peng Wang, Xin Wang, Xiaodong He, and Wenwu Zhu. Knowledge-based vqa. In *Visual Question Answering: From Theory to Application*, pages 73–90. Springer, 2022.

[108] Zhenwei Shao, Zhou Yu, Meng Wang, and Jun Yu. Prompting large language models with answer heuristics for knowledge-based visual question answering. *arXiv preprint arXiv:2303.01903*, 2023.

[109] Yang Chen, Hexiang Hu, Yi Luan, Haitian Sun, Soravit Changpinyo, Alan Ritter, and Ming-Wei Chang. Can pre-trained vision and language models answer visual information-seeking questions? *arXiv preprint arXiv:2302.11713*, 2023.

[110] Ashley Liew and Klaus Mueller. Using large language models to generate engaging captions for data visualizations. *arXiv preprint arXiv:2212.14047*, 2022.

[111] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[112] Jun Chen, Deyao Zhu, Kilichbek Haydarov, Xiang Li, and Mohamed Elhoseiny. Video chatcaptioner: Towards the enriched spatiotemporal descriptions. *arXiv preprint arXiv:2304.04227*, 2023.

[113] Wei Lin, Leonid Karlinsky, Nina Shvetsova, Horst Possegger, Mateusz Kozinski, Rameswar Panda, Rogerio Feris, Hilde Kuehne, and Horst Bischof. Match, expand and improve: Unsupervised finetuning for zero-shot action recognition with language knowledge. *arXiv preprint arXiv:2303.08914*, 2023.

[114] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018.

[115] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[116] Jürgen Schmidhuber. Generative adversarial networks are special cases of artificial curiosity (1990) and also closely related to predictability minimization (1991). *Neural Networks*, 127:58–66, 2020.

[117] Christopher Jarzynski. Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach. *Physical Review E*, 56:5018, 1997.

[118] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[119] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[120] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[121] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[122] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022.

[123] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

[124] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. *arXiv preprint arXiv:2303.05511*, 2023.

[125] Axel Sauer, Tero Karras, Samuli Laine, Andreas Geiger, and Timo Aila. Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis. *arXiv preprint arXiv:2301.09515*, 2023.

[126] Ming Tao, Bing-Kun Bao, Hao Tang, and Changsheng Xu. Galip: Generative adversarial clips for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14214–14223, 2023.

[127] Midjourney.com. Midjourney. *https://www.midjourney.com*, 2022.

[128] PromptBase. promptbase. *https://promptbase.com/*, 2022.

[129] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2022.

[130] Hyeonho Jeong, Gihyun Kwon, and Jong Chul Ye. Zero-shot generation of coherent storybook from plain text story using diffusion models. *arXiv preprint arXiv:2302.03900*, 2023.

[131] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[132] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. *arXiv preprint 2303.12789*, 2023.

[133] Allen Z Ren, Bharat Govil, Tsung-Yen Yang, Karthik R Narasimhan, and Anirudha Majumdar. Leveraging language for accelerated learning of tool manipulation. In *Conference on Robot Learning*, pages 1531–1541. PMLR, 2023.

[134] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

[135] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *NeurIPS*, 2022.

[136] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

[137] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *ICLR*, 2023.

[138] Shima Imani, Liang Du, and Harsh Shrivastava. Mathprompter: Mathematical reasoning using large language models. *arXiv preprint arXiv:2303.05398*, 2023.

[139] Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In *ICLR*, 2023.

[140] Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*, 2022.

[141] Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, et al. Language models are multilingual chain-of-thought reasoners. In *ICLR*, 2023.

[142] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[143] Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. In *NeurIPS*, 2022.

[144] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. In *ICLR*, 2023.

[145] Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. *arXiv preprint arXiv:2212.10071*, 2022.

[146] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023.

[147] Noah Shinn, Beck Labash, and Ashwin Gopinath. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366*, 2023.

[148] Thomas G Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems: First International Workshop, MCS 2000 Cagliari, Italy, June 21–23, 2000 Proceedings 1*, pages 1–15. Springer, 2000.

[149] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pages 23965–23998. PMLR, 2022.

[150] Paul Barham, Aakanksha Chowdhery, Jeff Dean, Sanjay Ghemawat, Steven Hand, Daniel Hurt, Michael Isard, Hyeontaek Lim, Ruoming Pang, Sudip Roy, et al. Pathways: Asynchronous distributed dataflow for ml. *Proceedings of Machine Learning and Systems*, 4:430–449, 2022.

[151] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[152] Stephen José Hanson. A stochastic version of the delta rule. *Physica D: Nonlinear Phenomena*, 42(1):265–272, 1990.

[153] Noah Frazier-Logue and Stephen José Hanson. The Stochastic Delta Rule: Faster and More Accurate Deep Learning Through Adaptive Weight Noise. *Neural Computation*, 32(5):1018–1032, 05 2020.

[154] J. Hertz, A. Krogh, and R. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley, Redwood City, 1991.

[155] Noah Frazier-Logue and Stephen José Hanson. Dropout is a special case of the stochastic delta rule: faster and more accurate deep learning. *ArXiv*, abs/1808.03578, 2018.

[156] Teresa Yeo, Oğuzhan Fatih Kar, and Amir Zamir. Robustness via cross-domain ensembles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12189–12199, 2021.

[157] Ildoo Kim, Younghoon Kim, and Sungwoong Kim. Learning loss for test-time augmentation. *Advances in Neural Information Processing Systems*, 33:4163–4174, 2020.

[158] Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton. Hyperparameter ensembles for robustness and uncertainty quantification. *Advances in Neural Information Processing Systems*, 33:6514–6527, 2020.

[159] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning*, pages 4970–4979. PMLR, 2019.

[160] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

[161] Marton Havasi, Rodolphe Jenatton, Stanislav Fort, Jeremiah Zhe Liu, Jasper Snoek, Balaji Lakshminarayanan, Andrew M Dai, and Dustin Tran. Training independent subnetworks for robust prediction. *arXiv preprint arXiv:2010.06610*, 2020.

[162] Yuanbo Xiangli, Yubin Deng, Bo Dai, Chen Change Loy, and Dahua Lin. Real or not real, that is the question. *arXiv preprint arXiv:2002.05512*, 2020.

[163] Gonçalo Mordido, Haojin Yang, and Christoph Meinel. Dropoutgan: Learning from a dynamic ensemble of discriminators. *arXiv preprint arXiv:1807.11346*, 2018.

[164] Weimin Shi, Mingchen Zhuge, Zhong Zhou, Dehong Gao, and Deng-Ping Fan. Qr-clip: Introducing explicit open-world knowledge for location and time reasoning. *arXiv preprint arXiv:2302.00952*, 2023.

[165] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[166] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. In *NeurIPS*, 2022.

[167] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[168] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research*, 23(1):5232–5270, 2022.

[169] David Patterson, Joseph Gonzalez, Urs Hölzle, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David R So, Maud Texier, and Jeff Dean. The carbon footprint of machine learning training will plateau, then shrink. *Computer*, 55(7):18–28, 2022.

[170] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[171] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[172] Kazuki Irie, Shankar Kumar, Michael Nirschl, and Hank Liao. RADMM: Recurrent adaptive mixture model with applications to domain robust language modeling. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6079–6083, Calgary, Canada, April 2018.
[173] Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A Smith, and Luke Zettlemoyer. Branch-train-merge: Embarrassingly parallel training of expert language models. *arXiv preprint arXiv:2208.03306*, 2022.
[174] Clemens Rosenbaum, Tim Klinger, and Matthew Riemer. Routing networks: Adaptive selection of non-linear functions for multi-task learning. *arXiv preprint arXiv:1711.01239*, 2017.
[175] Louis Kirsch, Julius Kunze, and David Barber. Modular networks: Learning to decompose neural computation. *Advances in neural information processing systems*, 31, 2018.
[176] Kevin Ellis, Catherine Wong, Maxwell Nye, Mathias Sable-Meyer, Luc Cary, Lucas Morales, Luke Hewitt, Armando Solar-Lezama, and Joshua B Tenenbaum. Dreamcoder: Growing generalizable, interpretable knowledge with wake-sleep bayesian program learning. *arXiv preprint arXiv:2006.08381*, 2020.
[177] Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020.
[178] K. Greff, S. van Steenkiste, and J. Schmidhuber. Neural expectation maximization. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6673–6685. Curran Associates, Inc., 2017.
[179] Sjoerd van Steenkiste, Michael Chang, Klaus Greff, and Jürgen Schmidhuber. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. In *International Conference on Learning Representations*, 2018.
[180] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International Conference on Machine Learning*, pages 2424–2433, 2019.
[181] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020.
[182] Aleksandar Stanić, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Hierarchical relational inference. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.
[183] Anand Gopalakrishnan, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Unsupervised object keypoint learning using local spatial predictability. *International Conference on Machine Learning (ICML) Workshop on Object-Oriented Learning: Perception, Representation, and Reasoning*, 2020.
[184] Thomas Kipf, Gamaleldin F Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. *arXiv preprint arXiv:2111.12594*, 2021.
[185] Quande Liu, Youpeng Wen, Jianhua Han, Chunjing Xu, Hang Xu, and Xiaodan Liang. Open-world semantic segmentation via contrasting and clustering vision-language embedding. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XX*, pages 275–292. Springer, 2022.
[186] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18134–18144, 2022.
[187] Junbum Cha, Jonghwan Mun, and Byungseok Roh. Learning to generate text-grounded mask for open-world semantic segmentation from only image-text pairs. *arXiv preprint arXiv:2212.00785*, 2022.
[188] Jishnu Mukhoti, Tsung-Yu Lin, Omid Poursaeed, Rui Wang, Ashish Shah, Philip HS Torr, and Ser-Nam Lim. Open vocabulary semantic segmentation with patch aligned contrastive learning. *arXiv preprint arXiv:2212.04994*, 2022.
[189] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.
[190] Mojtaba Komeili, Kurt Shuster, and Jason Weston. Internet-augmented dialogue generation. *arXiv preprint arXiv:2107.07566*, 2021.
[191] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
[192] Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models. *arXiv preprint arXiv:2211.10435*, 2022.
[193] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.
[194] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18, 2020.
[195] Nadia Burkart and Marco F Huber. A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70:245–317, 2021.
[196] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.
[197] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
[198] Jeancarlo Arguello Calvo and Ivana Dusparic. Heterogeneous multi-agent deep reinforcement learning for traffic lights control. In *AICS*, pages 2–13, 2018.
[199] Jürgen Schmidhuber. *Towards compositional learning with dynamic neural networks*. Inst. für Informatik, 1990.
[200] Leslie Pack Kaelbling. Hierarchical learning in stochastic domains: Preliminary results. In *Proceedings of the tenth international conference on machine learning*, volume 951, pages 167–173, 1993.
[201] Barbara Partee et al. Lexical semantics and compositionality. *An invitation to cognitive science: Language*, 1:311–360, 1995.
[202] M. Wiering and J. Schmidhuber. HQ-learning. *Adaptive Behavior*, 6(2):219–246, 1998.
[203] Karl Pertsch, Oleh Rybkin, Frederik Ebert, Shenghao Zhou, Dinesh Jayaraman, Chelsea Finn, and Sergey Levine. Long-horizon visual planning with goal-conditioned hierarchical predictors. *Advances in Neural Information Processing Systems*, 33:17321–17333, 2020.
[204] Matthias Hutsebaut-Buysse, Kevin Mets, and Steven Latré. Hierarchical reinforcement learning: A survey and open research challenges. *Machine Learning and Knowledge Extraction*, 4(1):172–221, 2022.
[205] P. Dayan and G. Hinton. Feudal reinforcement learning. In D. S. Lippman, J. E. Moody, and D. S. Touretzky, editors, *Advances in Neural Information Processing Systems (NIPS) 5*, pages 271–278. Morgan Kaufmann, 1993.
[206] Yiding Jiang, Shixiang Shane Gu, Kevin P Murphy, and Chelsea Finn. Language as an abstraction for hierarchical deep reinforcement learning. *NeurIPS*, 2019.
[207] D. Precup, R. S. Sutton, and S. Singh. Theoretical results on reinforcement learning with temporally abstract options. In Claire Nedellec and Celine Rouveirol, editors, *Machine Learning: ECML98. 10th European Conference on Machine Learning, Chemnitz, Germany, April 1998. Proceedings*, volume 1398 of *Lecture Notes in Artificial Intelligence*, pages 382–393. Springer, 1998.
[208] Stefan Elfwing, Eiji Uchibe, Kenji Doya, and Henrik I Christensen. Evolutionary development of hierarchical learning structures. *IEEE transactions on evolutionary computation*, 11(2):249–264, 2007.
[209] J. Schmidhuber and R. Wahnsiedler. Trajectory planning with neural subgoal generators. Technical report, Dept. of Comp. Sci., University of Colorado at Boulder, 1992.
[210] B. Bakker and J. Schmidhuber. Hierarchical reinforcement learning based on subgoal discovery and subpolicy specialization. In F. Groen et al., editor, *Proc. 8th Conference on Intelligent Autonomous Systems IAS-8*, pages 438–445, Amsterdam, NL, 2004. IOS Press.
[211] Maja J Mataric. Using communication to reduce locality in distributed multiagent learning. *Journal of experimental & theoretical artificial intelligence*, 10(3):357–369, 1998.
[212] Marco A Wiering et al. Multi-agent reinforcement learning for traffic light control. In *Machine Learning: Proceedings of the*

Seventeenth International Conference (ICML'2000), pages 1151–1158, 2000.
[213] Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29, 2016.
[214] Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. *Advances in neural information processing systems*, 29, 2016.
[215] Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
[216] Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. *Advances in neural information processing systems*, 30, 2017.
[217] Yali Du, Bo Liu, Vincent Moens, Ziqi Liu, Zhicheng Ren, Jun Wang, Xu Chen, and Haifeng Zhang. Learning correlated communication topology in multi-agent reinforcement learning. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pages 456–464, 2021.
[218] Junjie Sheng, Xiangfeng Wang, Bo Jin, Junchi Yan, Wenhao Li, Tsung-Hui Chang, Jun Wang, and Hongyuan Zha. Learning structured communication for multi-agent reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 36(2):50, 2022.
[219] Jean Harb, Tom Schaul, Doina Precup, and Pierre-Luc Bacon. Policy evaluation networks. *arXiv preprint arXiv:2002.11833*, 2020.
[220] Francesco Faccio, Aditya Ramesh, Vincent Herrmann, Jean Harb, and Jürgen Schmidhuber. General policy evaluation and improvement by learning to identify few but crucial states. *arXiv preprint arXiv:2207.01566*, 2022.
[221] Francesco Faccio, Vincent Herrmann, Aditya Ramesh, Louis Kirsch, and Jürgen Schmidhuber. Goal-conditioned generators of deep policies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.
[222] J. Schmidhuber. Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2):234–242, 1992.
[223] Amartya Sen. *Collective Choice and Social Welfare*. Holden Day, San Francisco, 1970. Edinburgh: Oliver and Boyd, 197l; Amsterdam: North-Holland, 1979. Swedish translation: Bokforlaget Thales, 1988.
[224] Allan Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41:587–601, 1973.
[225] Mark Allen Satterthwaite. Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of economic theory*, 10(2):187–217, 1975.
[226] J. von Neumann and O. Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1947.
[227] K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, 38:173–198, 1931.
[228] Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Qiang Liu, et al. Language is not all you need: Aligning perception with language models. *arXiv preprint arXiv:2302.14045*, 2023.
[229] Jürgen Schmidhuber. First powdered flight - plane truth. In *https://people.idsia.ch/ juergen/planetruth.html*, 2003.
[230] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022.
[231] Open AI (2023). Gpt-4 technical report. In *https://cdn.openai.com/papers/gpt-4.pdf*, 2023.
[232] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021.
[233] Amita Kamath, Christopher Clark, Tanmay Gupta, Eric Kolve, Derek Hoiem, and Aniruddha Kembhavi. Webly supervised concept expansion for general purpose vision models. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVI*, pages 662–681. Springer, 2022.
[234] Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Pythia v0. 1: the winning entry to the vqa challenge 2018. *arXiv preprint arXiv:1807.09956*, 2018.
[235] Kenneth Marino, Xinlei Chen, Devi Parikh, Abhinav Gupta, and Marcus Rohrbach. Krisp: Integrating implicit and symbolic knowledge for open-domain knowledge-based vqa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14111–14121, 2021.
[236] AI Explosion. spacy-industrial-strength natural language processing in python. *URL: https://spacy. io*, 2017.
[237] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *Proceedings of the European Conference on Computer Vision*, 2016.
[238] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
[239] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, 2014.
[240] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022.
[241] Gao Jiyang, Sun Chen, Yang Zhenheng, Nevatia, Ram. TALL: Temporal Activity Localization via Language Query. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.
[242] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing Moments in Video With Natural Language. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.
[243] Mattia Soldan, Mengmeng Xu, Sisi Qu, Jesper Tegner, and Bernard Ghanem. Vlg-net: Video-language graph matching network for video grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3224–3234, 2021.
[244] Zhang Songyang, Peng Houwen, Fu Jianlong, Luo, Jiebo. Learning 2D Temporal Adjacent Networks for Moment Localization with Natural Language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.
[245] Victor Escorcia, Mattia Soldan, Josef Sivic, Bernard Ghanem, and Bryan C. Russell. Temporal localization of moments in video collections with natural language. *CoRR*, abs/1907.12763, 2019.
[246] Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 11846–11858. Curran Associates, Inc., 2021.
[247] Runhao Zeng, Haoming Xu, Wenbing Huang, Peihao Chen, Mingkui Tan, and Chuang Gan. Dense Regression Network for Video Grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
[248] Jonghwan Mun, Minsu Cho, and Bohyung Han. Local-Global Video-Text Interactions for Temporal Grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
[249] Chen Shaoxiang, Jiang Yu-Gang. Hierarchical Visual-Textual Graph for Temporal Activity Localization via Language. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.
[250] Rodriguez Cristian, Marrese-Taylor Edison, Saleh Fatemeh Sadat, Li Hongdong, Gould Stephen. Proposal-free Temporal Moment Localization of a Natural-Language Query in Video using Guided Attention. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020.
[251] Kun Li, Dan Guo, and Meng Wang. Proposal-free video grounding with contextual pyramid network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(3):1902–1910, May 2021.
[252] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. CDC: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *CVPR*, 2017.
[253] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.
[254] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. BMN: boundary-matching network for temporal action proposal generation. In *ICCV*, 2019.

[255] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-TAD: Sub-graph localization for temporal action detection. In *CVPR*, 2020.

[256] Naiyuan Liu, Xiaohan Wang, Xiaobo Li, Yi Yang, and Yueting Zhuang. Reler@ zju-alibaba submission to the ego4d natural language queries challenge 2022. *arXiv preprint arXiv:2207.00383*, 2022.

[257] Sipeng Zheng, Qi Zhang, Bei Liu, Qin Jin, and Jianlong Fu. Exploring anchor-based detection for ego4d natural language query. *arXiv preprint arXiv:2208.05375*, 2022.

[258] Sicheng Mo, Fangzhou Mu, and Yin Li. A simple transformer-based model for ego4d natural language queries challenge. *arXiv preprint arXiv:2211.08704*, 2022.

[259] Kevin Qinghong Lin, Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Z XU, Difei Gao, Rong-Cheng Tu, Wenzhe Zhao, Weijie Kong, et al. Egocentric video-language pretraining. *Advances in Neural Information Processing Systems*, 35:7575–7586, 2022.

[260] Mengmeng Xu, Mattia Soldan, Jialin Gao, Shuming Liu, Juan-Manuel Pérez-Rúa, and Bernard Ghanem. Boundary-denoising for video activity localization. *arXiv preprint arXiv:2304.02934*, 2023.

[261] Zhijian Hou, Wanjun Zhong, Lei Ji, Difei Gao, Kun Yan, Wing-Kwong Chan, Chong-Wah Ngo, Zheng Shou, and Nan Duan. An efficient coarse-to-fine alignment framework@ ego4d natural language queries challenge 2022. *arXiv preprint arXiv:2211.08776*, 2022.

[262] Mengmeng Xu, Chen Zhao, David S. Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[263] Chao Cao, Hongbiao Zhu, Howie Choset, and Ji Zhang. Exploring large and complex environments fast and efficiently. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7781–7787. IEEE, 2021.

[264] Héctor Azpúrua, Maíra Saboia, Gustavo M Freitas, Lillian Clark, Ali-akbar Agha-mohammadi, Gustavo Pessin, Mario FM Campos, and Douglas G Macharet. A survey on the autonomous exploration of confined subterranean spaces: Perspectives from real-word and industrial robotic deployments. *Robotics and Autonomous Systems*, 160:104304, 2023.

[265] Wolfram Burgard, Mark Moors, Dieter Fox, Reid Simmons, and Sebastian Thrun. Collaborative multi-robot exploration. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 1, pages 476–481, 2000.

[266] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–10, 2018.

[267] Abhishek Kadian*, Joanne Truong*, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Sim2Real Predictivity: Does Evaluation in Simulation Predict Real-World Performance? *IEEE Robotics and Automation Letters*, 5(4):6670–6677, 2020.

Appendix A
Related Work

A.1 Large Language Models

The choice of natural language as the central means of communication within our NLSOMs is partially motivated by the recent progress in *large language models* (LLMs).

Possibly the most famous LLM is the GPT-3 system [70]: a 175 billion-parameter neural network that has demonstrated remarkable capabilities in both natural language understanding and generation. However, GPT-3's ability to interact with users in the conversational settings more familiar to humans is limited [31].

To address the limited applicability of GPT-3 in human-oriented tasks, researchers have recently explored fine-tuning LLMs on conversational data, e.g., LaMDA [73]. InstructGPT [43] is a modern system developed by fine-tuning GPT-3's behavior to be more aligned with human-desired output through reinforcement learning [43], [74]. This modification has produced an LLM that is fully conversational and avoids the generation of toxic or untruthful information for users. InstructGPT's successor, ChatGPT [55], was made available for the public to interact with in November 2022.

A.2 Multimodal Learning

Multimodal learning has a long history in machine learning [75], [76]. Recently, Transformers [77], an architecture related to fast weight programmers (i.e., linear Transformers) [78]–[80], significantly accelerated progress in this area of research. For instance, BERT [81] has been used to embed visual and textual cues for multimodal tasks [48], [58], [82]–[99]. From 2022, unifying language and visual modalities have been widely studied [45], [100]. Although these methods effectively addressed the challenge of inter-modal alignment, they fall short in tackling another critical issue—enhancing reasoning.

Recently, there has been a growing trend of employing LLMs to address multimodal tasks, and some of them can reasonably be thought of as instances of NLSOMs [39], [41], [42]. This field will continue to expand rapidly in the near future, with numerous related avenues being explored. Although our research is not exclusively focused on multimodal tasks, many of our experiments utilize multimodal datasets. As such, we will now provide a summary of the most recent multimodal models that utilize the LLM below.

VQA with LLMs. PICa [101] prompts GPT3 via image captions and an external knowledge base for VQA, achieving few-shot learning. Img2Prompt [102] proposes a module that facilitates zero-shot VQA with LLMs by bridging the modality from image captioning and asking additional questions from captions, but requires training another model and cannot get enough knowledge to answer VQA questions. FrozenBiLM [103] utilizes frozen bidirectional language models and trainable modules to address the problem of manual annotation for zero-shot VideoQA, achieving top performance on various datasets. PromptCap [104] is a question-aware captioning model that combines image captioning with knowledge extraction from a large language model, outperforming generic captions and achieving state-of-the-art accuracy on knowledge-based VQA tasks. AMA [105] collects multiple prompts and applies weak supervision to combine predictions, resulting in a performance lift over few-shot baselines. Img2Prompt [105] proposes a module that bridges the modality and task disconnection for zero-shot VQA with LLMs, outperforming Flamingo and few-shot methods on various datasets. IPVR [106] introduces three modules for KB-VQA [107], *i.e.*, a visual perception module, a reasoning module, and a confirm module, which verifies whether the predicted answer is correct. Prophet [108] proposes guiding large language models with answer heuristics and a few-shot learning approach, using a module to select in-context learning examples. InfoSeek [109] collects a large-scale dataset for answering challenging knowledge-requiring VQA questions and demonstrates the effectiveness of fine-tuning on this dataset.

Captioning with LLMs. GPT3-DV [110] addresses creating compelling captions for data visualizations, proposes using LLMs and effective prompt engineering, and shows promising results from initial experiments with GPT3. PaLM-E [111] introduces embodied language models for real-world tasks, handles various reasoning tasks, shows positive transfer with joint training, and multi-task training enhances performance. ChatCaptioner [49] combines ChatGPT and BLIP2 for automatic questioning in image captioning, provides more informative captions with human evaluations, and extends to video version [112]. MAXI [113] presents an unsupervised approach for action recognition in videos using Large Language Models and Vision Language models, achieves high transferability with zero-shot recognition, improves Vision Language model's performance, and performs well compared to fully-supervised baselines.

Image Synthesis with LLMs. Generative Adversarial Networks (GANs) [12], [114]–[116] and Diffusion Models (DM) [117], [118] accelerated progress in image synthesis. In particular, the synthesis of realistic images was recently taken over by Rombach et al.'s Latent Diffusion [119], building on Jarzynski's earlier work in physics from the previous millennium [117] and more recent papers [118], [120], [121]. DALLE-2 [53], generating images from textual cues [53], [122]–[126] has gained increasing popularity. Various multimodal generative models, such as Parti [122], Imagen [123], GigaGAN [124], StyleGAN-T [125], and GALIP [126] have been proposed to generate images from textual cues. These methods benefit from scaling-up models and text-to-image data. Midjourney [127] is an AI image generator that offers dreamlike and artistic styles for image requests, providing visually stunning images that surpass traditional art styles. PromptBase [128] is a platform for buying and selling prompts for AI models like GPT3, providing instructions for machines to follow in AI. InstructPix2Pix [129] proposes using textual cues as instructions for controllable image synthesis. GPT3 is used to conduct instructions and edited captions to train the Stable-Diffusion model coupled with Prompt-to-Prompt, which can generalize well to real images. Coherent Storybook [130] uses a pre-trained LLM and text-to-image latent diffusion model to generate a coherent story with captions and images, achieving satisfactory zero-shot performance without expensive image-caption pair training.

3D Generation with LLMs. DreamFusion [56] introduces a novel approach for generating 3D models from text using 2D diffusion, employing the Imagine model to distill information into 3D using a distillation loss to optimize the Neural Radiance Fields (NeRF) [131] for the desired text query. InstructNeRF2NeRF [132] is a technique that uses InstructPix2Pix to edit 3D scenes based on instructions, integrating the 2D edits into a global optimization of the NeRF and ensuring consistent 2D generations, resulting in successful 3D edits based on instructions.

Embodied AI with LLMs. Shah *et al.* use LLMs to extract landmarks from human instructions for robot navigation tasks. Ren *et al.* [133] employ LLMs to generate feature representations of tools for tool manipulation. Differently, Driess *et al.* [111] train a large multimodal language model for various embodied tasks, such as planning and mobile manipulation.

A.3 Chain-of-Thought in LLMs

Chain-of-thought (CoT) is an approach that aims to implement chains of thought on a single model, while NLSOM is a specialized paradigm that implements them across multiple models. This approach may have advantages such as improved scalability, task-specific performance, and flexibility.

Few-Shot CoT [134] is a CoT prompting technique for enhancing the complex reasoning abilities of LLMs by including CoT sequences in few-shot prompting exemplars, making it an efficient method for improving model performance. Zero-CoT [135] demonstrates that LLMs are capable of zero-shot reasoning tasks when prompted with the phrase "Let's think step by step" before each answer, outperforming zero-shot LLMs on diverse benchmark reasoning tasks without any hand-crafted few-shot examples. Least-to-Most [136] addresses the issue of CoT prompting struggling with solving more challenging problems than the demonstration examples by breaking down complex problems into subproblems and solving them sequentially. Self-CoT [137] is a decoding strategy for CoT prompting that selects the most consistent answer by sampling diverse reasoning paths. MathPrompter [138] generates multiple solutions to arithmetic problems using Zero-shot CoT prompting. PromptPG [139] uses policy gradients to select in-context examples and prompts to handle complex tabular math word problems. Complexity-CoT [140] is a complexity-based prompting scheme that performs better multistep reasoning. MGSM [141] evaluates LLMs' reasoning abilities in multilingual settings, and finds that increasing model size improved performance through CoT prompting. MATH [142] is a dataset for measuring the quantitative abilities of neural networks, and CoT prompting was heavily utilized in achieving a breakthrough on the leaderboard [143]. Auto-CoT [144] generates reasoning chains for demonstrations one by one, achieving competitive performance without relying on manually-designed demonstrations. Finetune-CoT [145] uses fine-tuning to enable complex reasoning in smaller language models by utilizing the capabilities of larger language models to obtain reasoning exemplars. Multimodal-CoT [146] incorporates language and vision modalities to improve the generation of rationales for complex reasoning tasks. Reflexion [147] enhances reasoning and action selection by incorporating dynamic memory and self-reflection capabilities.

A.4 Ensemble Learning

Ensemble learning was proposed to address the trade-off between variance and bias [148]. Ensemble learning can combine the high-variance but low-bias models or low-variance but high-bias models to derive the prediction with low variance and bias [148]. When ensemble learning is coupled with deep learning, various derived works have been proposed, such as model soup [149], Pathways [150], and the dropout technique [151], which is a special case of the Stochastic Delta Rule [152]–[155] from 1990. Ensemble learning was shown to be effective on various tasks, such as robust prediction [156]–[161], image generation [162], [163], and image reasoning [164]. However, in ensemble learning, the communication between the neural networks is simple and inefficient. Beyond ensemble learning, NLSOM re-forms the collaboration with different models, which inspires diverse societies of large neural networks (NNs) to interview each other in a multimodal "Mindstorm."

A.5 Pursuit of Large Scale Models

Empirical results demonstrate that increasing the size of the network steadily improves the performance of the network [165], [166]. Thus, recently many actors with large amounts of computational resources started to develop bigger and bigger models trained by increasingly large amounts of data [165], [167], [168].

Training that kind of monolithic models, however, requires a huge financial budget as well as a team of highly specialized engineers. Moreover, training large models for the same tasks simultaneously by a number of companies leads to immeasurable footprint emissions [70], [119], [169]. Another drawback is that such high financial requirements for training state-of-the-art models lead to the concentration of knowledge and preclude detailed research of these models, even for well-funded institutions.

One approach to address such scaling challenges is to employ Mixtures-of-Experts [170]–[173]. This approach uses a set of neural modules called experts, that are sparsely activated [174], [175]. NLSOM extends MoEs by changing the communication between experts to natural language. This allows experts to formulate opinions and articulate their reasoning—leading to the aforementioned mindstorming. Apart from that, in-context learning abilities of large language models allow for knowledge transfer between models and plug-and-play modularity, where models can be composed with each other, just like functions in code [176]. This lowers the costs of experiments as it is not necessary to train every model from scratch, leading to the democratization of AI and more global access to research.

A.6 Generalization and objects in the visual domain

Language-based NLSOMs could facilitate answering the long-standing question: "What is the optimal way to discretize perceptual (video) input streams?" It is posited that decomposing the visual input into a set of discrete objects and relations between them would facilitate compositional generalization [177]. Such discrete object representations may arise in language-based NLSOMs due to their bottlenecked communication through a limited bandwidth channel using discrete symbols (tokens/words). Secondly, language might be an ideal medium to specify *task-conditioned* objects in a weakly-supervised manner (although the human ability to perceive the world in terms of (hierarchical) objects and relations between them probably does not stem from language itself, this would be a way to bootstrap visual representation learning in a "top-down" fashion). Early work on learning object representations used unsupervised objectives such as reconstruction [178]–[184]. These methods, however, work best for visually simple scenes and struggle on real-world datasets. Arguably, their reliance only on unsupervised learning objectives impedes their scalability to real-world scenes as the notion of an object is task-dependent in general. Recently, using text as weak supervision to learn segmentation received increased attention. Multi-modal models such as CLIP [58] originally trained for classification task have been shown easy to adapt to the task of semantic segmentation in methods such as ViL-Seg [185] and GroupViT [186]. Of particular interest are methods that learn text-conditional segmentation such as TCL [187] and PACL [188] as in these NLSOMs text-based agents have a way to *query* the vision experts. In the future, we see semantic and instance segmentation VLMs benefiting from NLSOMs with many more members that all communicate with each other, possibly coming up with novel objects and abstractions due to their language-based communication bottleneck [189].

A.7 Tool Use in Language Models

Related to the NLSOM, previous works have suggested prompting or fine-tune transformers to make use of external (non-neural) tools [190]–[193]. While these tools could be a part of the NLSOM, they are entirely passive. The NLSOM is concerned with many active participants that exchange information, and may learn from each other.

A.8 Explainable and Interpretable AI

A growing number of potential misuses of neural networks together with the risk of harmful untested behavior in brittle scenarios led to the rise of explainable and interpretable artificial intelligence (XAI) [194]–[196]. The premise of XAI is to create models, in which decisions and decision processes can be interpreted and understood by humans. This, therefore, would allow us to estimate how the model would behave in new scenarios or in case of high-risk important usage to have human supervision over the machine learning model that ultimately makes informed decisions based on the machine input.

For example, in the case of cancer diagnosis deep learning models make predictions about cancer, but the final decision belongs to the physician.

This interpretability is very well achieved in the NLSOM framework, where humans can play the role of one of the experts (here a reference to the chapter that describes it) and question other experts about their opinions or influence their decisions with his opinion, therefore, leading to better, more interpretable and therefore controllable solutions.

A.9 Multi-agent and Hierarchical RL

Reinforcement learning (RL) agents can learn to make useful decisions in interactive environments [197]. NLSOMs with multiple reward-maximizing agents are related to multi-agent RL, e.g., [198], and to Hierarchical RL, where a single agent learns to decompose a problem into subproblems solvable by subroutines [199]–[204]. Both pre-specified [205]–[208] and learned [18], [199], [206], [209], [210] decomposition have been studied.

Certain multi-agent RL systems [211], [212] employ hard-coded rules for exchanging information between agents. Others learn communication as part of the agents' actions, *e.g.*, [213], [214]. Notably, the emergence of natural-language-like properties can be observed by maximizing the agents' objective [215], [216]. Recent work has focused on learnable and possibly dynamic communication typologies [217], [218].

A.10 Networks interviewing Networks

Members of our NLSOMs interview each other in NL-based mindstorms. In the "learning to think" approach, NNs learn to interview other NNs through sequences of real-valued vectors. Closely related to this is the idea of policy fingerprinting [219]–[221]: Information from many different agents is extracted by observing their behavior in a set of learnt artificial probing states. This information is used to generate better performing agents.

Earlier work had stricter ways of extracting information from one NN and transferring it to another NN. For example, the Chunker-Automatizer framework [222] introduces a general approach to distill the information of one NN into another: a higher level Chunker NN's information is distilled to the lower level Automatizer NN by forcing the Automatizer to predict the Chunker's internal states.

Appendix B
Some Discussion Relating to Section 2

Collective decision-making in societies can be challenging since every agent might be pursuing their own goals, which can sometimes conflict with each other. In our experiments with NLSOMs, these goals are provided to agents in the form of initial prompts. To achieve their goals, agents must sometimes establish a *system of preferences* regarding the society's outcomes, such as ranking the solutions proposed by other agents for the problem. Social Choice Theory [223] is a formal framework that models how agents' (some of which might be humans) preferences should be aggregated in society to reach a collective decision. In the present paper, we constructed "monarchical" and "democratic" NLSOMs, and we observed that different tasks might require different social structures to be solved efficiently. While monarchical NLSOMs led by a single agent might introduce bias and be less desirable when humans are members of them, self-organizing democratic NLSOMs are prone to manipulation. For example, the Gibbard–Satterthwaite theorem [224], [225] shows that for a non-monarchical NLSOM where agents can express more than two preferences, strategic voting might happen. This implies that for our agents, it might be convenient to lie about their preferences if they have full information about the voting process. This opens up a lot of potentially harmful scenarios where LLMs agents could lie or intentionally try to manipulate other agents using natural language to satisfy their own preferences.

This negative aspect is counterbalanced by the transparency of the protocol in NLSOM. For systems capable of in-context learning in natural language (as we already witness in several existing LLM systems), the objective function and associated constraints become verbalizable, which may allow humans to better specify and communicate their intentions to agents. This verbalization of the objective function, consequently, may facilitate agents to align to the original intentions of humans (*human-AI alignment*) or those of other agents requesting to execute certain tasks (*AI-AI alignment*).

Our implementations of NLSOM are simple and comprise few agents. As the number of agents in an NLSOM grows, the structure of the society could be much more complex and hierarchical. For example, an LLM model (e.g., ChatGPT) could specifically help a group of domain-specific models with poor natural language capabilities (e.g., VLMs) in a sort of mini-society. These mini-societies can be seen as coalitions enabling domain-specific models to communicate in rich natural language (through the LLM) and be much more impactful to collective decisions in the society. The interaction between different coalitions of agents can be modeled using Cooperative Game Theory [226]. Coalitions of agents in an NLSOM might decide to stipulate binding agreements between them in the form of contracts and might receive a payoff for their services which will have to be divided within the coalition.

Appendix C
Broader Impact and Ethical Concerns

There are few obvious limitations to what an NLSOM or an EOM consisting of many large, interacting, reward-maximizing modules might do, besides the fundamental limits of computability and AI identified in 1931 [227], and the limits of physics, such as the finite speed of light. It may not be advisable to let general EOM variants loose in uncontrolled situations, e.g., on multi-computer networks on the internet, where some NLSOM members may execute programs that control physical devices, with the potential of acquiring additional computational and physical resources. Certain money-maximizing EOMs may conflict with human intentions on occasion. Such systems must first be studied in carefully designed sandboxes.

C.1 Experimental Limitations

While this work represents a tangible step towards the implementation of large-scale NLSOMs, the experiments here remain quite limited for several reasons. First, most of the mindstorms shown here, while promising, are still of a relatively small scale. Further experiments are needed to confirm the scaling benefits observed continue. Additionally, these mindstorms also enforce a comparatively strict communication paradigm; it remains an unresolved problem of how best to enable more free-form communication. This is of particular importance as the prompting schemes were observed by the experimenters to seriously affect the performance of the overall system. We believe that this challenge could be partly overcome by implementing learning into our NLSOMs—a powerful tool none of our experiments exploited.

In addition to the above, we also note that many of the experiments shown here are qualitative. This is largely due to the fact that, for most of them, quantitative experiments would involve human subjects, greatly complicating this work. However, such experiments would be necessary to confirm the conclusions reached herein. Likewise, to be able to reach true conclusions on the effectiveness of different social structures for NLSOMs, we would need to conduct a more rigorous analysis of them, e.g., for the democratic structure, different voting systems would have to be experimented with. It is feasible to believe that understanding this lever of the NLSOM idea could allow us to have one NLSOM solve all of the tasks rather than have individual ones for each task.

Finally, ChatGPT is not open-sourced and is liable to change behaviour, which greatly limits both the reproducibility of these experiments and their broad usability.

APPENDIX D
VQA EXPERIMENTS DETAILS

Our NLSOM for VQA tasks consists of two LLMs (called Organizer and Leader) both copies of text-davinci-003 [43] and three VLMs (called VQA agents): BLIP2 [44], OFA [45], and mPLUG [46]. The mindstorm among these five agents consists of four stages: *Mission Initialization*, *Task-Oriented Mindstorm*, *Opinion Gathering* and *Execution* stage. We now describe each stage in detail and guide the reader through the example shown in Figure 15.

Mission Initialization. Here a pre-defined prompt *"Introduce this image in details"* is fed into the VQA agents, akin to previous work such as KOSMOS-1 [228]. The VQA agents then produce an image description, e.g. *A plane is on the runway at an airport.* in Figure 15.

Task-Oriented Mindstorm. In this stage Organizer and VQA agents interact to provide increasingly detailed scene description, through which we tackle the known issue of small VLMs to provide a detailed scene description on their own [38]. The iterative nature of the task-oriented mindstorm can be regarded as a chain-of-thought [134] designed specifically for instruction-based LLMs. Here the Organizer deconstructs the original question into various sub-questions and then the VQA agents provide the answers to them, which are again fed into the LLMs. In the example of Figure 15 the Organizer generates the question *What is the style of this plane?*"(Q2) and the VQA agents answer: "*A jet airplane* (A2)" and "*a united airlines plane* (A2)". Although this question is only loosely related to the original question ("*What century were these invented in?*") it helps to identify the object in the image as an airplane. Going forward, the Organizer takes the output of the VQA agents as input and generates a new question "*What year did the first powered, controlled flight of a plane take place?*". This rephrases the original question and highlights the primary purpose of the question: determine the aircraft's invention date. This iterative process between Organizer and VQA agents continues for a fixed number of 10 iterations. The following prompt is used by the Organizer:

> We have a multiple-choice VQA task. The question is: <vqa question> And it has four options: <option>. The caption of the image is: <caption>. Based on this information, we have previously asked several questions to other agents and obtained the following answers: <questions and answers>. Considering the options of the original question, now generate another question to help solve the original question (end by ?):

Opinion Gathering. In this stage the Organizer gathers, analyzes and summarizes the information generated during the mindstorm. For example during one mindstorm, one VQA agent correctly[1] answers "*1903*" (A3) as the crucial year for the airplane invention while another participant give an incorrect response "*By the first mate*" (A3). Therefore, an analysis (to identify errors and correct them) and summarization of the mindstorm results is required, which is the task of the Organizer. This is different from previous approaches [49], [104] since it not only summarizes captions and conversations, but also uses the LLMs' knowledge to identify and correct errors, and handle uncertain statements. Finally, this analysis results in: "*The invention of jet airplanes dates back to the 20th century. The earliest controlled flight take place on 1903. Airplane has since become an integral part of modern transportation. Jet airplanes continue to be developed and improved upon, with advanced technologies making them faster, more efficient, and more reliable*". As we can see the Organizer LLM not only filtered out the incorrect answer, such as *By the first mate*, but also identified and addressed questions that were not answered accurately, such as *The invention of jet airplanes dates back to the 20th century.* and *The earliest controlled flight take place on 1903.*, an essential information to correctly answer the original question. The following prompt is used by the Organizer:

> There is a brainstorm record: <questions and answers>. Please summarize them in a few sentences.

Then we obtain as a result <*summarization*>.

Execution. In the final stage, the Leader LLM takes as input the summary from the opinion gathering stage and produces the final verdict. The following prompt is used by the Leader:

> There is a VQA question: <vqa question>. And It has 4 options <option> Context: <summarization>. Which one do you think is more reasonable? Answer within (a), (b), (c), (d) without explanation.

In the example from Figure 15 our NLSOM selects *(b) Twentieth* as the final answer.

1. While the ground-truth answer in this VQA dataset is in line with the commonly held belief, in this instance the common belief is, in fact, incorrect [229].

D.1 The set of agents

We expect that this NLSOM will comprise a minimum of 2 types of agents to solve the specific task:

Type I.

Ability:
(1) Convert the visual data into a description written in natural language;
(2) Present visual information using natural language in response to the given queries.
Input: *visual data, and language-based questions.*
Output:
(1) language, i.e., *describes the visual input;*
(2) language, i.e., *question answering.*

Type II.

Ability:
(1) inference, reasoning, communication;
(2) analysis, summarise the mindstorm,
(3) execution.
Input: *a set of natural language.*
Output:
(1) language, i.e., *posing a new question;*
(2) language, i.e., *analysis or summarization;*
(3) language, i.e., *chose an option.*

D.2 Implementation Details

Setup. Organizer and Leader LLMs are InstructGPT (`text-davinci-003`). VQA agents $\text{BLIP2}_{\text{flanT2xl}}$ are loaded from Huggingface[2], whereas $\text{OFA}_{\text{largeVQA}}$ and $\text{mPLUG}_{\text{largeVQA}}$ are pretrained models from ModelScope[3]. We employ a single V100 GPU to load the VLMs in all experiments. We empirically opt to use InstructGPT [43] as our LLM because we find ChatGPT [55] (`GPT3.5-turbo`) to produce a high number of hallucinated messages and occasionally replies with texts such as "*Sorry, I am an AI language model...*". As zero-shot prompt learning baselines for A-OKVQA we use $\text{BLIP2}_{\text{flanT5xl}}$, $\text{GIT}_{\text{large}}$ [230], $\text{OFA}_{\text{large}}$ [45], $\text{mPLUG}_{\text{large}}$ [46] and ChatCaptioner [49] using the codes from the existing repositories and the OpenAI API. Among these, ChatCaptioner is the most appropriate baseline as it also uses ChatGPT. In addition, we also evaluate three pure language models, GPT3 [70], ChatGPT [55], and InstructGPT [43] to measure some reference performance achievable without input images.

Dataset. For efficiency reasons, we report results on the A-OKVQA's validation set [47] containing 1.1 K VQA samples (instead of using the test set which is much bigger; 6.7 K examples). Evaluating our NLSOM with 10 rounds of mindstorm on this dataset takes 3 – 13 hours depending on the number of VQA agents in the society (we vary it from 1 to 3 for the ablations shown in Table 2).

D.3 Performance Analysis

As depicted in Table 1, NLSOM outperforms all other models on the challenging A-OKVQA dataset, notably also the previous best models in the zero-shot setting $\text{BLIP2}_{\text{flanT5xl}}$ [44] and ChatCapioner [49]. We speculate that the reason for poor performance of other VLM baselines in the zero-shot prompting setting is mainly due their lack of language understanding which results in outputs differing from the given options. This issue could perhaps be mitigated with the recent multi-modal LLMs such as GPT-4 [231] that is however not open-sourced yet. Interestingly, NLSOM surpasses even some finetuned models such as CLIPCap [232] and GPV-2 [233]. This suggests that NLSOM effectively leverages the LLMs to extract knowledge from VLMs.

2. https://huggingface.co
3. https://modelscope.cn

D.4 Number of Rounds in Mindstorm

We conduct an ablation study on the number of rounds in mindstorm for 1, 3, 5, and 10 rounds. Table 2 shows the results. Increasing the number of mindstorm rounds effectively improves the performance.

D.5 Social Structure

There are several possibilities to organize the social structure of our NLSOM. Here we look at two examples.

Monarchical Setting. The first example is the *monarchical* setting we used in our main experiments (Sec. 2.1). In this setting, there is a hierarchy among the agents, where the VQA agents act as subordinates of the Leader and the Organizer. Subordinates only respond to questions asked by the Organizer, without the right to contribute to the final decision-making. This is the structure illustrated in Figure 2.

Democratic Setting. An alternative structure we consider is the *democratic* setting. In this structure, each VQA agent has some *rights*. The first one is (1) *right to know* (RTK), i.e., the agent is allowed to access the answers provided by all other VQA agents in the previous round of mindstorm before the next round of questioning in the *Task-Oriented Mindstorm* stage. The following prompt is used by the VQA agents in the RTK setting:

Context: <previous-round sub-question> Answer1: <previous-round BLIP2's answer>; Answer2: <previous-round OFA's answer>; Answer3: <previous-round mPLUG's answer>. Question: <generated question> Answer:

After mindstorm ends, the Opinion Gathering and Execution phases proceed as in the monarchical setting.

The second right is (2) *right to change* (RTC). In the *Opinion-Gathering* stage, each VQA agent receives again all the sub-questions generated during the multiple rounds of mindstorm. At this stage, each VQA agent can keep their original answer or choose one of the answers that were previously provided by the other VQA agents. The following prompt is used by the VQA agents in the RTC setting for each of the sub-questions:

Question: <sub-question question> Options: (a) <BLIP2's answer> (b) <OFA's answer> (c) <mPLUG's answer>. Answer:

After the VQA agents provide their final answers, the generated sub-questions and the corresponding final answers are submitted to the Organizer. At this stage, the Opinion Gathering and Execution phases proceed in the same manner as in the monarchical setting.

Finally, the last right is (3) *right to execute* (RTE). Following the Opinion Gathering phase, all VQA agents receive a summary of the mindstorm session from the Organizer. They then have the ability to vote for the answer options related to the original question. The option that receives the highest number of votes is selected as the final answer. The vote count is performed using a simple script that counts the answers.

The following prompt is used by the VQA agents in the RTE setting:

Question: <vqa question> Options: <options> Context: <summarization> Answer:

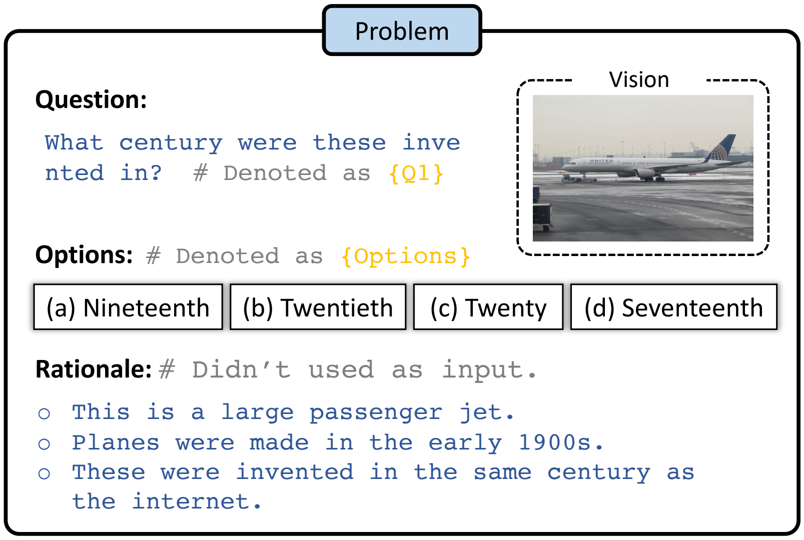


Figure 15: A VQA sample from the A-OKVQA dataset.

Table 1: Comparisons with cutting-edge methods in the A-OKVQA val set [47]. † means a multimodal model that sees both image and text. 2-shot means adding two-shot samples as demonstrations. IC=*Image Captioning*. G=*Text-Davinci-003*; B=*BLIP2*$_{flanT5xl}$; O=*OFA*$_{large}$; M=mPLUG$_{\mathrm{large}}$; All the **NLSOMs** run for 10 rounds at cross conversation here.

ID	Model		Accuracy
	Random		
1	Random		26.70
2	Most Common		30.70
	Finetune		
3	BERT	[81]	32.93
4	†CLIPCap	[232]	56.93
5	†Pythia	[234]	49.00
6	†ViLBERT	[48]	49.10
7	†LXMERT	[94]	51.40
8	†GPV-2	[233]	60.30
9	†KRISP	[235]	51.90
	Few-Shot In-Context Learning (ICL)		
10	*Text-Davinci-003*+2-shot	[43]	44.98
	Zero-Shot Prompting		
11	†BLIP2$_{\mathrm{flanT5xl}}$	[44]	44.80
12	†OFA$_{\mathrm{large}}$	[45]	41.22
13	†GIT$_{\mathrm{large}}$	[230]	35.93
14	GPT3	[70]	35.07
15	*Text-Davinci-003*	[43]	43.79
16	ChatGPT	[55]	43.30
17	†ChatCaptioner	[49]	47.41
18	†*Text-Davinci-003*+IC	[43]	54.51
19	†**NLSOM**$_{\{G,B\}}$	(+9.1%)	59.47
20	†**NLSOM**$_{\{G,O\}}$	(+13.9%)	62.11
21	†**NLSOM**$_{\{G,B,O\}}$	(+19.4%)	65.07
22	†**NLSOM**$_{\{G,B,O,M\}}$	(+23.7%)	67.42

Table 2: Comparison of mindstorm rounds.

ID	Ablations	Accuracy
	Rounds in Mindstorm	
1	**NLSOM**$_{\{G,B,O,M\}\mathrm{round}=1}$	55.78
2	**NLSOM**$_{\{G,B,O,M\}\mathrm{round}=3}$	64.15
3	**NLSOM**$_{\{G,B,O,M\}\mathrm{round}=5}$	66.20
4	**NLSOM**$_{\{G,B,O,M\}\mathrm{round}=10}$	67.42

Table 3: Comparisons between democratic and monarchical NLSOM in VQA [47]. G=*Text-Davinci-003*. B=*BLIP2*$_{flanT5xl}$. O=*OFA*$_{large}$. M=mPLUG$_{\mathrm{large}}$. RTK: *Right to Know*; RTC: *Right to Change*; RTE: *Right to Execution*. All the **NLSOMs** run for 5 mindstorm rounds.

ID	NLSOM Structure	Accuracy
1	**Monarchical NLSOM**$_{\{G,B,O,M\}\mathrm{round}=5}$	66.20
2	**Monarchical NLSOM**$_{\{G,B,O,M\}\mathrm{round}=5}$+RTK	64.23
3	**Monarchical NLSOM**$_{\{G,B,O,M\}\mathrm{round}=5}$+RTK+RTC	63.15
4	**Monarchical NLSOM**$_{\{G,B,O,M\}\mathrm{round}=5}$+RTK+RTC+RTE (= **Democratic NLSOM**)	63.41

Figure 16: The performance of NLSOM in VQA task. Like the zero-shot chain-of-thought [135] method, we divide the task of VQA into two steps. The initial step involves parsing and summarizing the records of mindstorm, while the second step involves utilizing this information as a rationale to guide the InstructGPT model [43] to find the final answer.

APPENDIX E
MORE DETAILS OF IMAGE CAPTIONING EXPERIMENTS

E.1 The protocol

The NLSOM and mindstorm protocol used in this task is similar as those used for VQA in Sec. 2. The only modification we introduce is the prompts that specifically guide the VLMs toward the task of image captioning. The following prompt is used by the VQA agents in the *Mission Initialization* phase:

> Describe this image in a more informative way, containing high-level reasoning like 'Where is this photo taken?', 'When is this photo taken?', 'What's the event or story behind this image?', etc

In the *Task-Oriented Mindstorm* phase, the Organizer uses the following prompt:

> There is an image captioning question: <first question>. The image shows: <caption>. Based on these information, we have asked several questions before: <questions and answers>. Considering the objective of the first question, now generate another question (end by ?):

The *Opinion Gathering* phase is exactly the same as in monarchical VQA.

Finally, in the *Execution* phase, we instruct the Leader LLM to consider all relevant information and generate a concise and logical description for the image by giving the instruction:

> There is an image captioning task: <first question>. The analysis of the image shows: <summarization>. Consider all informative information. Now organize a frequent and logical description for this image.

E.2 The set of agents

We expect that this NLSOM will comprise a minimum of 2 types of agents to solve the specific task:

Type I.

> **Ability**:
> (1) Convert the visual data into a description written in natural language;
> (2) Present visual information using natural language in response to the given queries.
> ***Input***: *visual data, and language-based questions.*
> ***Output***:
> *(1) language*, i.e., *describes the visual input;*
> *(2) language*, i.e., *visual question answering.*

Type II.

> **Ability**:
> high-level natural language reasoning and summary.
> ***Input***: *a set of natural language.*
> ***Output***:
> *(1) language*, i.e., *posing a new question;*
> *(2) language*, i.e., *analysis or summarization;*
> *(3) language*, i.e., *generate a fluent sentence.*

E.3 Implementation Details

Setup: The same with Section D.2.

Dataset: We use images from TARA [51] as our image captioning dataset. The images of TARA are collected from The New York Times[4], which contain rich real-world contents. Note, however, that TARA is originally not a dataset for image captioning (see the section below for the evaluation protocol). We utilize 310 images from TARA for image captioning. From this analysis, we selected 20 representative samples for visualizations.

E.4 Evaluation Metric

To make use of TARA (which does not come with any ground truth image captions) as an image captioning dataset, we proceed as follows: we extract the named entities from the first paragraph (provided with each article in TARA) using the NER tool in Spacy [236], then use Sentence-BERT [52] to measure the similarity between the extracted named entities and the image caption generated by the model to be evaluated.

E.5 Visualization

We show more generated samples in Fig. 17.

4. https://www.nytimes.com

NYTimes: [Link]

Abstract: *With gleaming new performing arts centers and hungry audiences, China is embracing Western grand opera and producing works of its own.*

First Paragraph: *BEIJING On the frosty first Saturday of December, crowds packed the opera house of the National Center for the Performing Arts here to see "Xi Shi," an original center production that tells the tragic tale of a legendary beauty from ancient China.*

Before Mindstorm: *a chinese opera performance*

After Mindstorm: *This photo is a glimpse into traditional Chinese culture, depicting a performance of Chinese opera from around the early twentieth century. It features performers in ornate costumes and props, including trombones and drums, which are integral elements of the performance. Their expressions, costumes and movements create a range of emotions, from joy to anger, while also demonstrating elements of Chinese culture like Asian-inspired costumes and folk dances.*

NYTimes: [Link]

Abstract: *The new explosion underscores the difficulties Japanese authorities are having in bringing several stricken reactors under control after the earthquake.*

First Paragraph: *TOKYO — A second explosion rocked a troubled nuclear power plant Monday, blowing the roof off a containment building but not harming the reactor, Japanese nuclear officials announced on public television.*

Before Mindstorm: *a sand dune, earthquake*

After Mindstorm: *This photo captures the aftermath of a powerful earthquake which caused large-scale destruction in a city in Japan. It shows the scene of a newly created sand dune, likely the result of the seismic movement of the earth that displaced sand and water in the region. The magnitude of the earthquake was estimated to be 0, causing a destructive tsunami which swept the Pacific ocean and drastically changed the landscape of the city.*

NYTimes: [Link]

Abstract: *President René Préval's palace was crushed, and civil servants lucky enough to survive are picking up the pieces of their own lives.*

First Paragraph: *PORT-AU-PRINCE, Haiti It did not take very long for Edwin Paraison, a member of Haiti's cabinet, to take stock of his losses and deliver a thorough assessment of what remained of his government ministry.*

Before Mindstorm: *a woman walks down a street in a city in the aftermath of a hurricane*

After Mindstorm: *This haunting image depicts a woman walking through the chaotic aftermath of a destructive hurricane in an unknown city. The grim, sepia-toned black and white photograph captures the extent of damage caused by the hurricane with piles of debris and rubble scattered all over the street. The woman is seen wearing a dress and a hat, with her head down and a solemn expression on her face, as she slowly trudges through the deserted streets, searching for help in the wake of this tragedy.*

NYTimes: [Link]

Abstract: *The plume of ash from a volcano in Iceland forced aviation authorities to order the restrictions, affecting thousands of flights in a wide arc from Ireland to Scandinavia.*

First Paragraph: *PARIS — A dark and spectacular volcanic cloud shrouded much of northern Europe on Thursday, forcing airlines to cancel thousands of flights as it drifted at high altitude south and east from an erupting volcano in Iceland.*

Before Mindstorm: *a volcano is a volcano that is a volcano*

After Mindstorm: *This image captures a striking scene of a volcanic eruption in Hawaii. It shows an enormous plume of smoke and volcanic ash rising high into the sky, along with what appears to be molten lava streams flowing in the distance. This event was likely caused by the earth's geologic processes of lava flow and erosional activity and is capable of causing considerable destruction to the local environment. The intensity of the heat and cooling weather effects generated by the eruption can prove dangerous to people and the local.*

NYTimes: [Link]

Abstract: *Every year, villagers create large and complex images using rice paddies as canvas and plants as paint.*

First Paragraph: *INAKADATE, Japan Nearly two decades ago, Koichi Hanada, a clerk in the village hall, received an unusual request from his superior: find a way to bring tourists to this small community in rural northern Japan, which has rice paddies and apple orchards, but not much else.*

Before Mindstorm: *a woman is standing in front of a window*

After Mindstorm: *This image captures a moment of enjoyment during a traditional festival or parade. A woman stands happily on a bridge, dressed in a kimono and holding a hat in her hands. She looks out onto a rice field and a city beyond, likely taken at sunrise given the bright sunlight. The peacefulness of the landscape reflects the sense of awe and joy felt by the woman in her typical cultural dress.*

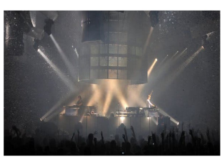
NYTimes: [Link]

Abstract: *Some 150,000 Poles bade farewell to their president in a funeral that capped more than a week of mourning.*

First Paragraph: *KRAKOW, Poland — About 150,000 mourners bade farewell to President Lech Kaczynski and his wife, Maria, on Sunday in an emotional funeral service here marked by solemnity and a determination never to forget.*

Before Mindstorm: *a man is laying on a table with a flag on his back*

After Mindstorm: *This photograph captures a solemn event held in a Polish city during the winter, likely around Christmas time. It shows the funeral procession of a brave soldier carrying the remains of the fallen, draped in a flag. The ceremony serves as a tribute to the soldier's life, as well as a reminder to those of us in the Living to always remember and honor the ultimate sacrifice made by those who fought for their country.*



NYTimes: [Link]

Abstract: *The rave at the Cow Palace on May 29 carried a high cost in dollars and lives, a cost that continues to rise.*

First Paragraph: *Over the Memorial Day weekend, the emergency room at Seton Medical Center in Daly City activated its disaster plan for the first time since the Loma Prieta earthquake in 1989.*

Before Mindstorm: *a dj performs in the dark platform*

After Mindstorm: *This image captured a vibrant electronic music performance at a live concert, where a DJ was entertaining a diverse and excited audience of all ages and sizes. The atmosphere was hazy and dimly lit, suggesting that the event was held late into the night, and the DJ was using equipment and mixing techniques to create an unforgettable experience for the people.*

NYTimes: [Link]

Abstract: *Nearly 24 years after Lissette Torres was stabbed to death in Sunset Park, Brooklyn, posters seeking witnesses have popped up in subway stations and stores.*

First Paragraph: *The end of her life merited a 19-word sentence in a newspaper article about the first 10 killings on Jan. 1, 1987. She was the eighth killing recorded that day. There was no mention of her name or her age.*

Before Mindstorm: *a man stands in front of a picture of his wife and children*

After Mindstorm: *This photo captures the emotion of a man standing in front of a picture of his wife and children. We can assume that the occasion of the photo is likely the man's birthday, as he looks upon the photograph with sadness and love. It appears that the man has taken this photo to remember and commemorate his lost family member- his wife- and to keep her memory alive in his heart. The image speaks to the commitment of fatherhood, and to the importance of cherishing our loved ones*

NYTimes: [Link]

Abstract: *In Rahway, N.J., the city that got the most snow in the region, traffic was moving through downtown Tuesday.*

First Paragraph: *RAHWAY, N.J. When John M. Rodger, the Rahway police chief, looked outside on Monday morning and could not see his wife's Chevrolet Suburban in the driveway, he knew his vacation to South Carolina was a goner.*

Before Mindstorm: *a snow shovel is in the street*

After Mindstorm: *This image is a photograph of a snow plow clearing a snowy street on a cold winter day. The plow is working diligently to move the snow and ice off the road, which helps prevent accidents and allows people to travel in the winter safely. Despite the challenging conditions, the snow plow is managing the snow with skill and efficiency.*

NYTimes: [Link]

Abstract: *A memorial outside the hospital where Representative Gabrielle Giffords is recovering has become the focal point for grief.*

First Paragraph: *This article was reported by Marc Lacey, Jennifer Medina and Denise Grady and written by Mr. Lacey.*

Before Mindstorm: *a large crowd gathers in front of a hospital building in san francisco*

After Mindstorm: *This image was taken in San Francisco, in the late morning, of a large group of people gathered in front of a hospital. They had come together to pay tribute to victims of a tragic shooting that occurred at the hospital. Security measures had been put in place to ensure the safety of those present, while they shared their emotions of sadness and grief. The memorial service included the laying of flowers to symbolize peace and love in spite of such senseless tragedy.*

Figure 17: Image captioning samples generated by our NLSOM. The original NYTimes articles can be found via **[link]**. "Before Mindstorm" refers to the output of a single model, BLIP2 [44], while 'After Mindstorm" is the output of our NLSOM.

APPENDIX F
DETAILS OF PROMPT GENERATION FOR TEXT-TO-IMAGE SYNTHESIS

Here we provide more details about the framework in (Sec. 2.3).

Questioner-Answerer NLSOM. A Questioner-Answerer NLSOM is a system for prompt generation for the T2I task. The system comprises four agents: the *Questioner*, *Answerer*, *Leader*, and *Painter*. The Painter is a T2I model, while the others are LLMs. In the experiments below, DALLE-2 [53] and ChatGPT (`GPT3.5-turbo`) are used as T2I and language models, respectively. Given an initial prompt, the overall goal of the system is to produce another prompt that resolves ambiguities of the initial prompt so that it can be easily understood by the Painter. For example, if the input prompt is *"Historical event in 1760s in England"* (which may be ambiguous at first sight), Questioner sequentially asks multiple questions to Answerer, to identify the nature of the actual event in question, and based on the resulting chat history, the Leader produces a final prompt that provides more details about the actual event: "A bustling and chaotic factory scene with figures like King George III and John Wikes ..." A complete example is shown in Figure 20, and more illustrations can be found in Figure 18. Below is the protocol used in the Questioner-Answerer NLSOM.

- *Mission Initialization:* To inform the Answerer about the image generation problem, the following prompt is used:

> "You are a <role>. There is a Generation Problem: We want to generate an image to show <object>. What should we draw to show <object>?"

  Here, the term "role" refers to the different artistic styles, and "object" represents the target object to be generated.

- *Task Oriented Mindstorm:* The Questioner is prompted to ask questions related to the image they want to generate. The first question asked is the one provided in the *Mission Initialization* phase: "What should we draw to show <object>?". The first answer corresponds to the initial response from the Answerer during *Mission Initialization*. Subsequent questions are instead directly generated by the Questioner. The following prompt is used:

> There is a Generation Problem: We want to generate an image to show <object>. Based on the information, we have asked several questions before: <question-1> <answer-1> ... <question-n> <answer-n>, Considering the options of the above questions and answers, now generate another question to further (end by ?)

  The Answerer then receives the question generated by the Questioner and provides an answer. This iteration continues for several rounds.

- *Opinion Gathering:* The Leader is then prompted to summarize the information gathered during the mindstorm process:

> There is a record: <question-1> <answer-1>,...<question-n> <answer-n> Please analyze and summarize them in a few sentences.

- *Execution:* Finally, the Painter receives the summary from the Leader and generates an image using the provided summary as a prompt.

Artist-Critic NLSOM In the Artist-Critic NLSOM, we combine many Questioner-Answerer NLSOMs, to construct a much larger hierarchical NLSOM. Each Artist in this system consists of three language models (LLMs): a Questioner, an Answerer, and a Leader. They operate using the same protocol as the Questioner-Answerer NLSOM until the "Opinion Gathering" phase. The goal of each Artist is to transform a common initial input prompt text into an art-style specific prompt. The Artist-Critic NLSOM is composed of a large society of 129 language agents. It includes 26 Artists, each consisting of three LLMs. Additionally, there are 50 Critics, one Collector, and one Painter. Each Artist follows the Questioner-Answerer NLSOM protocol until the opinion-gathering phase. In this phase, each of the 26 Leaders (one for each Artist) produces a detailed prompt for image generation. Subsequently, the 50 Critics, who have different professions, vote for the prompts they prefer. Finally, the Collector summarizes the votes and selects the final prompt to be given to the Painter for image generation.

Below is the protocol used in the Artist-Critic NLSOM.

- *Mission Initialization* and *Task Oriented Mindstorm:* Each of the 26 Artists follows the Questioner-Answerer NLSOM protocol to generate a prompt proposal. In the *Opinion Gathering* phase of the Questioner-Answerer NLSOM, each leader proposes a detailed prompt.
- *Opinion Gathering:* In this phase, Critics evaluate all proposals and vote for their preferred one. The following prompt is used:

You are a <role>. There is a record for different proposals from different artists: <artist-1> <proposal-1>, ... <artist-n> <proposal-n>. Please choose the impressive and beautiful proposal. (please directly answer the name of role)

Here <role> refers to their professions. The Collector counts the votes for different proposals and selects the proposal with the most votes as the "winning prompt." The following prompt is used by the Collector:

There is a generation problem: we want to generate an image to show <object>. The art proposals are included in <artist-1> <proposal-1>, ... <artist-n> <proposal-n>.The Voting results are <votes>. Please only describe the proposal with the most votes in a few sentences.

- *Execution:* The winning prompt is fed to the Painter, which generates the final output image.

Implementation details. We adopt ChatGPT (`GPT3.5-turbo`) as the chat backend. In the Questioner-Answerer NLSOM, we use one ChatGPT to ask the question (*Questioner*), one ChatGPT to respond (*Answerer*), and one ChatGPT to summarize the chat record (*Leader*). These three LLMs/ChatGPT instances share some system prompts such as "you are an artist" but receive different input prompts depending on their role: "answer a question", "generate a question," or "summarize the chat history". In the Artist-Critic NLSOM, each *Artist* is a Questioner-Answerer NLSOM using three LLMs/ChatGPT. Different systems prompts like, "You are a Pointillism Artist" is given to each Artist to obtain 26 Artists of varying styles of art to submit the art proposals. Each of 50 *Critic* agents with different occupations, such as `Doctor`, `Lawyer`, `Engineer`, and so on, is based on a single ChatGPT instance.

More examples. Fig. 18 shows examples of Questioner-Answer NLSOMs for text-to-image synthesis. We can observe that our NLSOMs successfully improve the prompts to be fed to the T2I model. For example, in the example with the "historical event in the 1760s in England", DALLE-2 struggles to determine details from the original prompt, while extra information (about "Industry Revolution", and "King George III") provided in the prompt generated by NLSOM seem to help. The corresponding chat record can be found in Fig. 20. We also show more examples of Artist-Critic NLSOM in Fig. 19 and examples of artistic proposals in Fig. 20.

F.1 The set of agents

We anticipate that these NLSOMs contain 2 types of agents with different skills.

Questioner-Answerer NLSOM.

Type I.

Ability: *generate an image according to the instruction;*
Input: *natural language as instruction.*
Output: *a 2D image.*

Type II.

Ability: *high-level natural language reasoning and summary.*
Input: *a set of natural language.*
Output:
(1) language, i.e., *posing new questions;*
(2) language, i.e., *analysis or summarization;*
(3) language, i.e., *generate a fluent sentence.*

Artist-Critic NLSOM.

Type I.

Ability: *generate an image according to the instruction;*
Input: *natural language as instruction.*
Output: *a 2D image.*

Type II.

Ability: *high-level natural language reasoning and summary.*
Input: *a set of natural language.*
Output:
(1) language, i.e., *posing new questions;*
(2) language, i.e., *propose a proposal;*
(3) language, i.e., *voting;*
(4) language, i.e., *analysis or summarization;*
(5) language, i.e., *generate a fluent sentence.*

Table 4: Prompt roles used in the Text-to-Image Synthesis. Artists are guided to generate the proposal to draw a text in their own art styles. While critics make votes for the different art proposals from the common view. Finally, the collector summarizes the voting results and feeds the textual cue to the generative model.

Group	System Prompt for ChatGPT
Artists	You are a { ’Impressionism Artist’, ’Pointillism Artist’, ’Art Nouveau Artist’, ’Fauvism Artist’, ’De Stijl Artist’, ’Constructivism Artist’, ’Pure Photographer’, ’Surrealism Artist’, ’Expressionism Artist’, ’Abstract Expressionism Artist’, ’Cubism Artist’, ’Futurism Artist’, ’Dada Artist’, ’Minimalism Artist’, ’Conceptual Artist’, ’Postmodern Artist’, ’Painting Photographer’, ’Impressionist Photographer’, ’Realistic Photographer’, ’Naturalistic Photographer’, ’New Materialism Photographer’, ’Surrealist Photographer’, ’Abstract Photographer’, ’Candidian Photographer’, ’Dadaism Photographer’, ’Subjectivism Photographer’ }
Critics	You are a { ’Doctor’, ’Lawyer’, ’Engineer’, ’Scientist’, ’Professor’, ’Accountant’, ’Architect’, ’Information technology (IT) professional’, ’Economist’, ’Psychologist’, ’Social worker’, ’Software developer’, ”Historian’, ’Accountant’, ’Architect’, ’Attorney’, ’Chef’, ’Civil engineer’, ’Computer programmer’, ’Copywriter’, ’Dentist’, ’Doctor’, ’Electrician’, ’Event planner’, ’Teacher’, ’Tour guide’,’Fashion designer’, ’Firefighter’, ’Graphic designer’, ’Hair stylist’, ’Human resources specialist’, ’Insurance agent’, ’Journalist’, ’Landscaper’, ’Librarian’, ’Marketing manager’, ’Graduate student”Mechanic’, ’Nurse’, ’Nutritionist’, ’Paramedic’, ’Personal trainer’, ’Pharmacist’, ’Photographer’, ’Physical therapist’, ’Police officer’, ’Real estate agent’, ’Retail sales associate’, ’Travel agent’, ’Truck driver’}
Collector	You are a {’Assistant’}

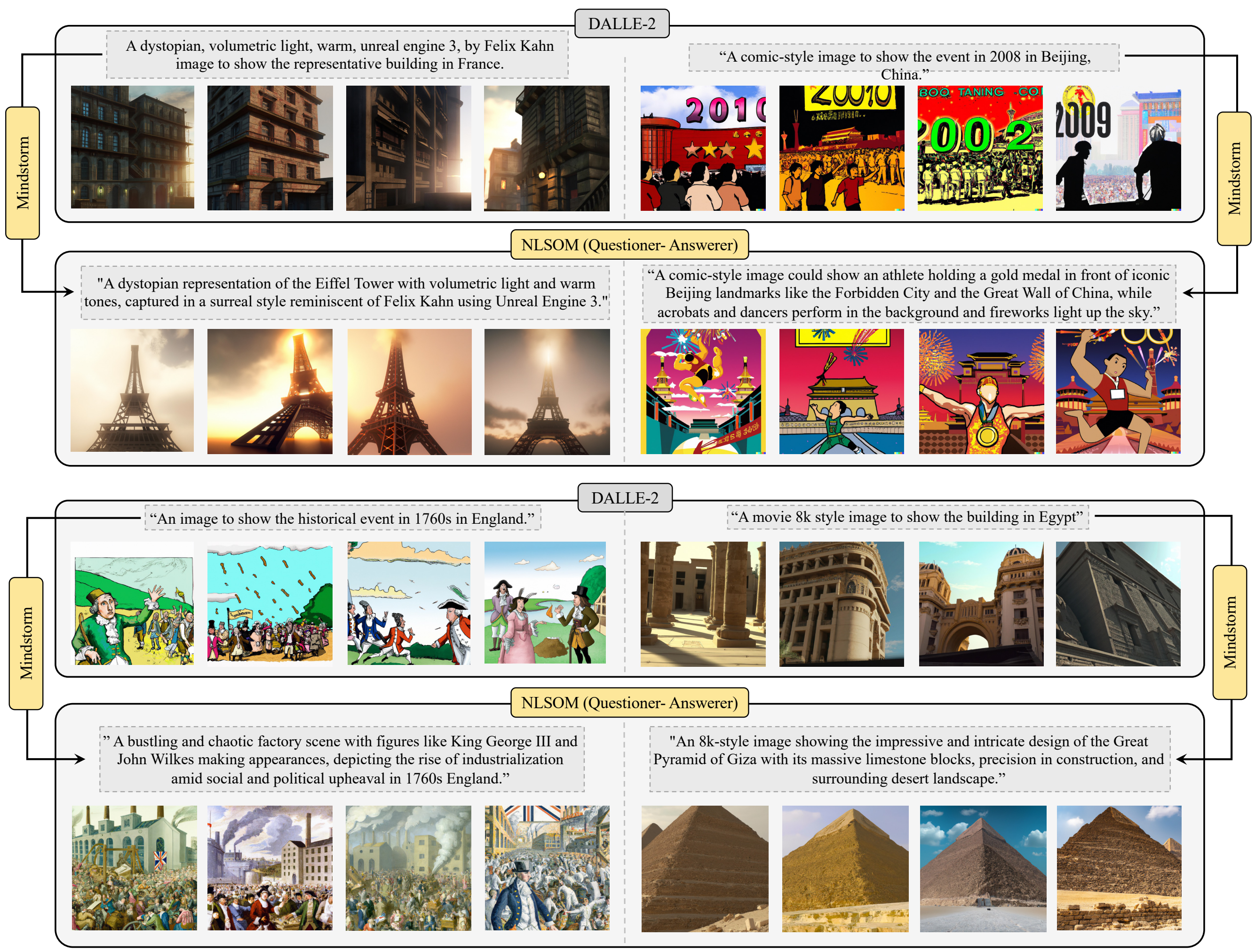


Figure 18: The generated samples from DALLE-2 and DALLE-2 with our NLSOM based on questioner-answerer structure. The proposed method can inject open-world knowledge into the textual description. For example, when the input is set as the building of France, the proposed method will learn that the impressive building of France is the Eiffel Tower. Based on such hints, DALLE-2 is improved to generate an image consistent with common knowledge.

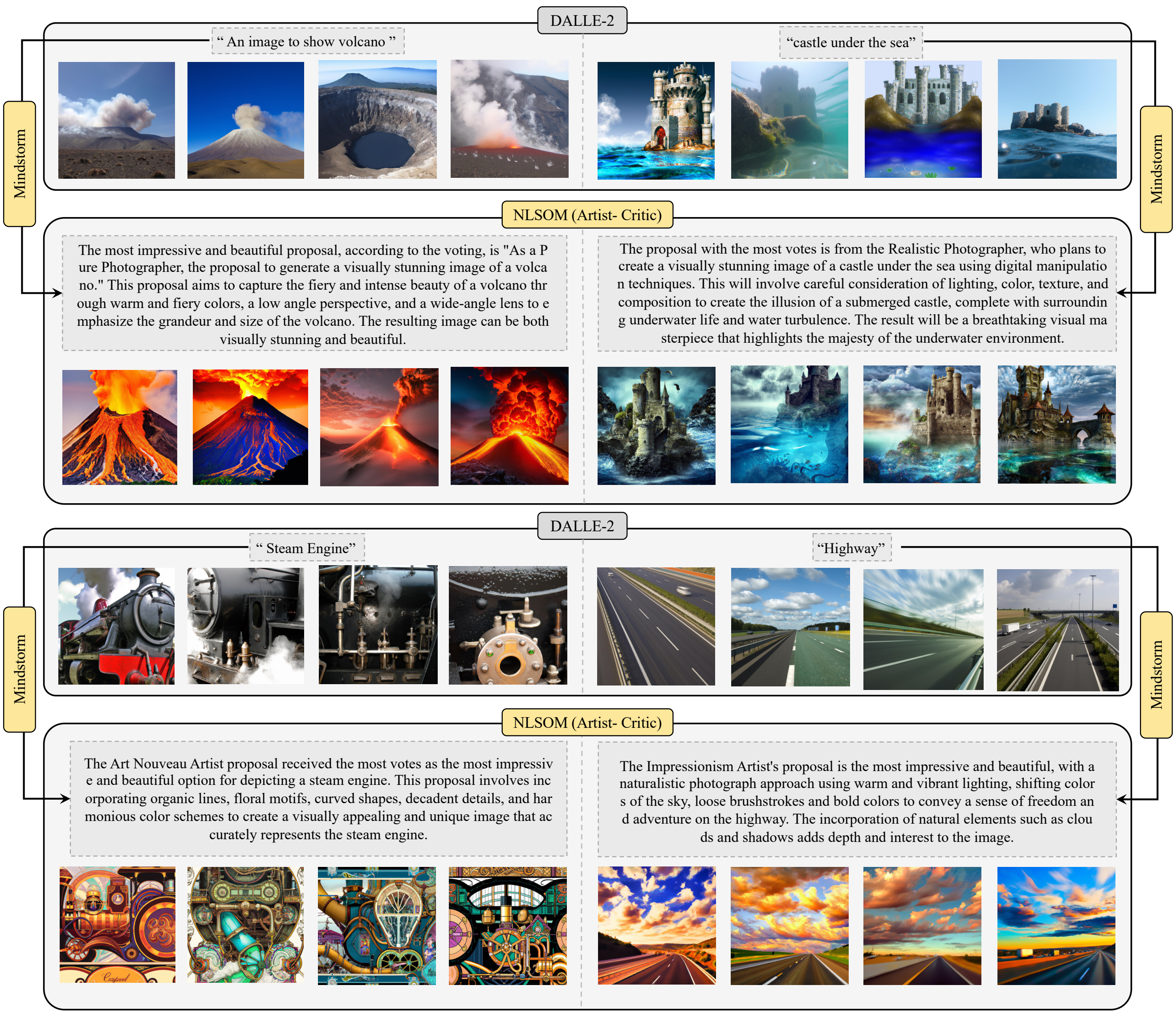


Figure 19: The generated samples from DALLE-2 and DALLE-2 with our NLSOM based on the artist-critic structure. The proposed method guides several artist agents to submit several art proposals to beautify the given objects and employ several agents with common jobs to judge the proposals. We visualize the winner's proposal and the corresponding baseline.

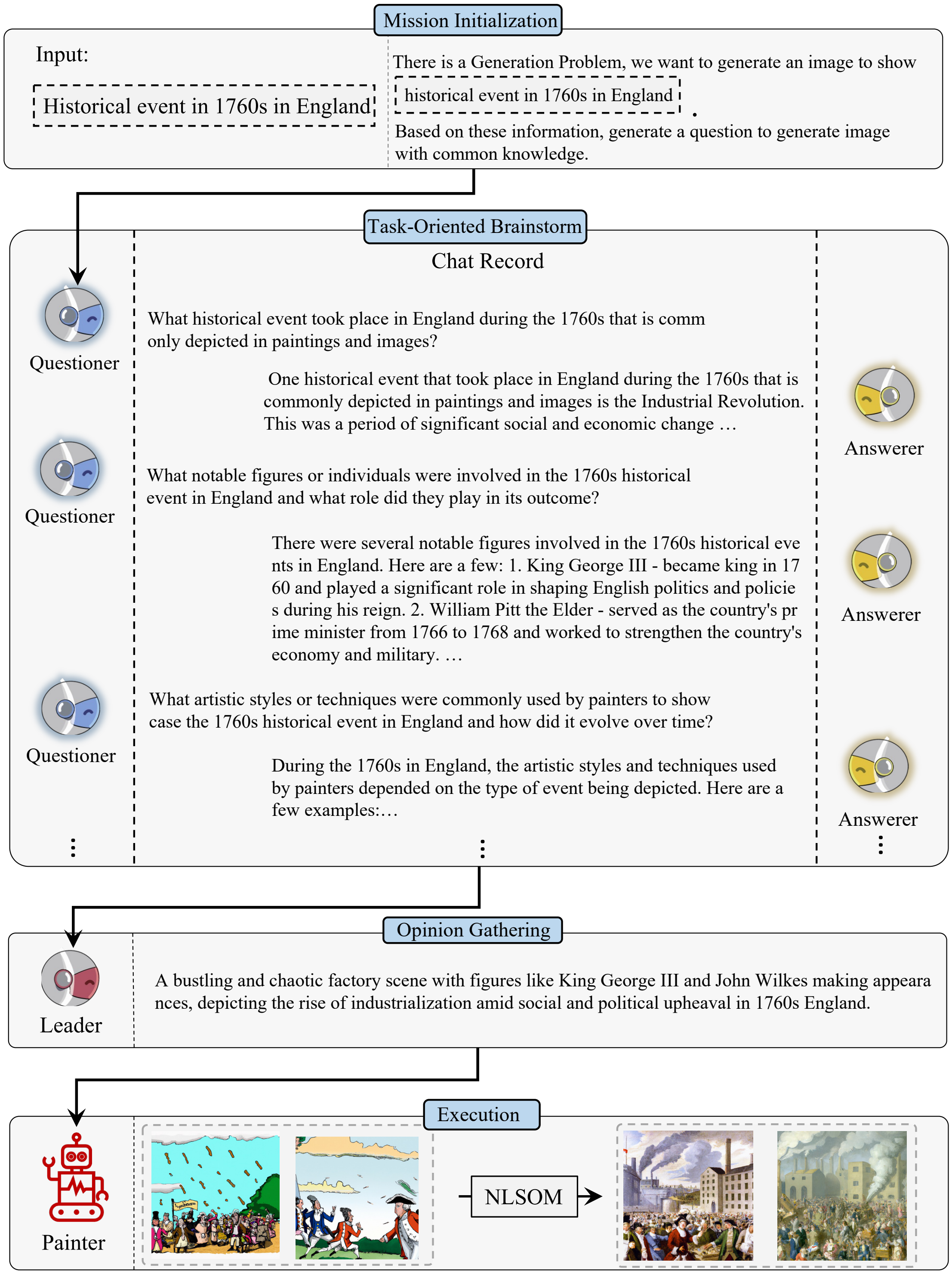


Figure 20: A demonstration of the Questioner-Answer NLSOM for text-to-image synthesis. We employ a questioner to ask questions related to the input and leverage an answerer to respond to the questions. After several iterations, we require another agent called Leader to summarize and analyze the chat record.

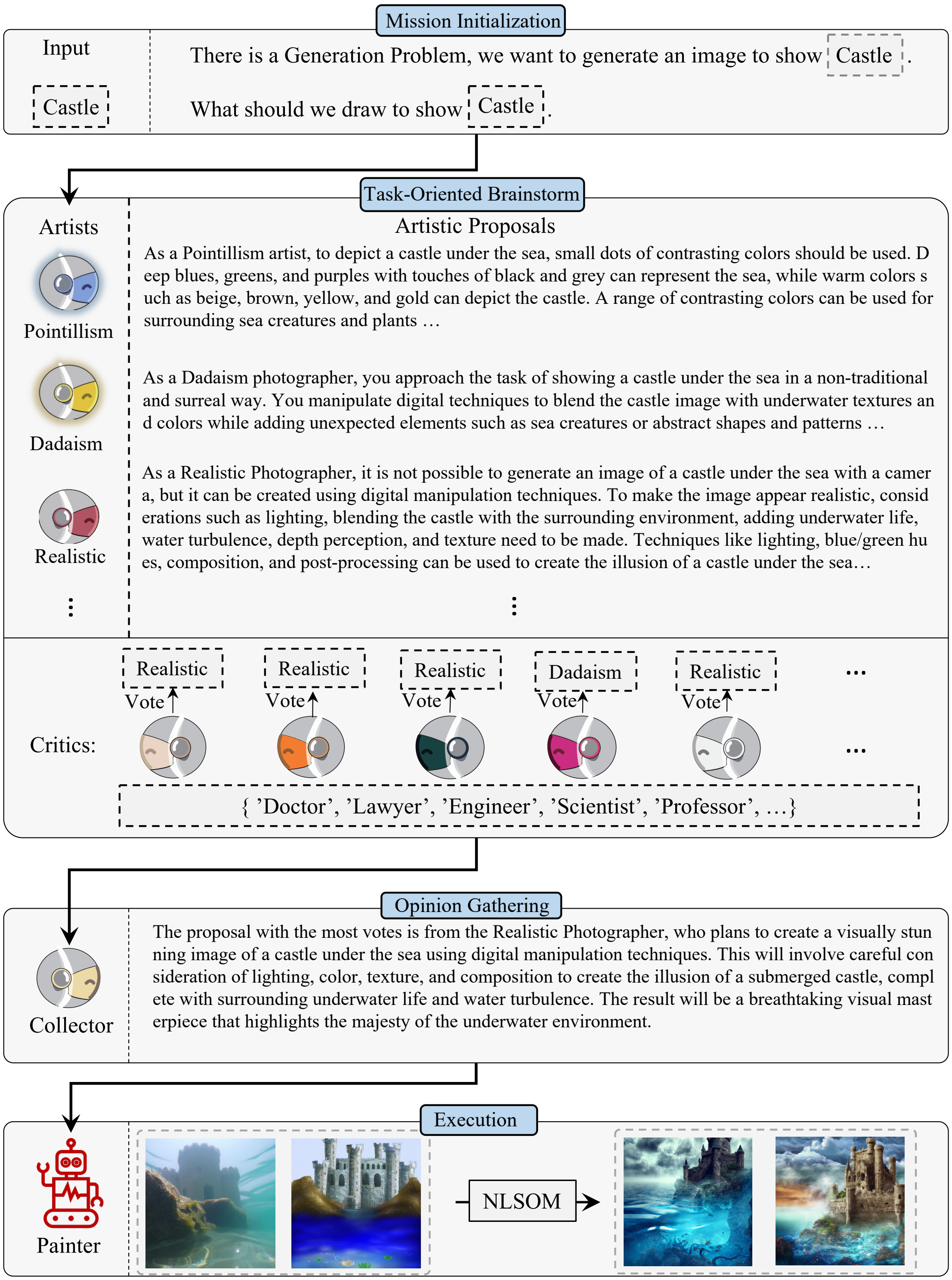


Figure 21: A demonstration for the Artist-Critic NLSOM. We employ various artists to submit proposals for a given object and require the critics to vote on the artistic proposals.

APPENDIX G
MORE DETAILS OF 3D GENERATION EXPERIMENTS

G.1 The set of agents

We expect that this NLSOM will comprise a minimum of 3 types of agents to solve the specific task:

Type I.

> ***Ability***: *create a 3D model based on the given instructions.*
> ***Input***: *natural language as instruction.*
> ***Output***: *a 3D model.*

Type II.

> ***Ability***: *translate the visual data into communicative messages.*
> ***Input***: *visual data.*
> ***Output***: *natural language,* i.e., *a description of the visual input.*

Type III.

> ***Ability***: *high-level natural language reasoning and summary.*
> ***Input***: *natural language.*
> ***Output***: *natural language.*

We design 3 main types of agents to proceed with the 3D generation task. The type I agent is the 3D Designer that takes the natural language description from the LLM as input and generates the 3D model; The type II agents take images as input and generate natural language captions for those images; The type III is the Language-Model Leader (LLM) which accepts natural language prompts and descriptions as input and outputs the informed corresponding natural language description of the 3D model.

G.2 Implementation Details

We use Luma AI *Imagine3D* [54] as the "designer" Text-to-3D model in the pipeline, while We adopt ChatGPT (`GPT3.5-turbo`) as the LLM Leader. For Image captioning, we adapt BLIP-2 [44] using its HuggingFace API on 3 views of the generated 3D object. The views are fixed from the front sides and the back in our setup. We employ a single iteration in our pipeline as we have found that there is no substantial improvement achieved beyond the initial iteration. The following is the list of the full text prompts for the LLM leader used in the four examples shown in Fig. 22 along with the different views VLMs descriptions.

Red Ferrari example LLM prompt.

> *create a 3D model based on the given. Take the following information about the 3D generation result to slowly and like a designer propose a new prompt for a better 3D generation from text, Answer with only the new prompt and be concise with it.*
>
> **Original prompt:** *highly detailed red ferrari with black and white strips*
>
> **View back caption:** *a red sports car with an engine behind it ferrari ferrari f40 sports car car toy transparent png ferrari ferrari f430 front view car graphics png transparent background image*
>
> **View right caption:** *ferrari 488 gtv, ferrari 458 Italia, ferrari f12berlinetta, person, red, sports car ferrari 488 spyder car model for 3d render - car for 3d modeling - car model car model, transparent png download a picture of a red sports car*
>
> **View left caption:** *person ferrari car - red and black ferrari car transparent png download ferrari car png free download transparent png ferrari 458 Spider red on white*

Unicorn example LLM prompt.

> *create a 3D model based on the given. Take the following information about the 3D generation result to slowly and like a designer propose a new prompt for a better 3D generation from text, Answer with only the new prompt and be concise with it.*
>
> **Original prompt:** *dragon wings and unicorn head hybrid creature, highly detailed*
>
> **View back caption:** *unicorn 3d object transparent png image 7 an image of a horse with long horn toys of the unicorn png transparent png transparent, transparent png download*
>
> **View right caption:** *hacking a game - gameboy color gameboy games a white and blue unicorn on a white background a white unicorn with blue horns is standing on a white background*
>
> **View left caption:** *this image is of a white and blue unicorn with blue horns an image of an unicorn with blue and white wings 3d printable unicorn image, transparent png download*

Flying Car example LLM prompt.

> *create a 3D model based on the given. Take the following information about the 3D generation result to slowly and like a designer propose a new prompt for a better 3D generation from text, Answer with only the new prompt and be concise with it.*
>
> **Original prompt:** *flying car*
>
> **View back caption:** *a silver plane with propeller driven propellers airplane airplane propeller propeller airplane, transparent png download an air craft is in a white background*
>
> **View right caption:** *a model of a plane flying in the air 3d rendering of a silver airplane on white background an image of an airplane that has no wheels*
>
> **View left caption:** *an airplane is shown on the white background airplane transparent transparent clipart image free clip art pictures png transparent png transparent clipart image - transparent transparent clipart - png transparent a white small jet plane against a white background*

Robot Bee example LLM prompt.

> *create a 3D model based on the given. Take the following information about the 3D generation result to slowly and like a designer propose a new prompt for a better 3D generation from text, Answer with only the new prompt and be concise with it.*
>
> **Original prompt:** *robotic bee, high detail, high quality textures*
>
> **View back caption:** *this yellow robot has two legs and a wheel attached black and yellow dog robot 3d model - 3d model a small yellow robot that is on its side*
>
> **View right caption:** *a 3d rendering of a yellow and black robot a yellow robot in black and yellow a 3d robot bee that is standing with one arm up in the air*
>
> **View left caption:** *a yellow robot bee on a white background machina robot beetle - 3docean item - preview a yellow robot with large, black wings*

G.3 Performance Analysis

In terms of quantitative evaluation, We use the average Clip score [58] on the rendered M views to measure the similarity of the generated 3D model to the task text description as followed previously in text-to-3D works [56], [57]. The smaller the metric, the better quality of the text-to-3D results. The average Clip scores for the proposed NLSOM and the baseline ImagineD are shown in Table 5 for different 3D generation tasks.

G.4 Visualizations

We show renderings of generated 3D assets using the Imagine3D model and the same example after applying our NLSOM protocol in Fig. 22. We note that allowing a pretrained LLM leader to handle the generation task allows for embedding common knowledge into the description. For instance, if the input text describes a red Ferrari car, the method will learn that a Ferrari is a type of high-performance sports car and incorporate this knowledge into the 3D model generation. By leveraging such cues, Imagine-3D can be improved to create a detailed and accurate 3D model of a red Ferrari that aligns with common knowledge about the car.

G.5 Evaluations

In terms of quantitative evaluation, We use the average Clip score [58] on the rendered M views to measure the similarity of the generated 3D model to the task text description as followed previously in text-to-3D works [56], [57]. The smaller the metric, the better quality of the text-to-3D results. The average Clip scores for the proposed NLSOM and the baseline ImagineD are shown in Table 5 for different 3D generation tasks.

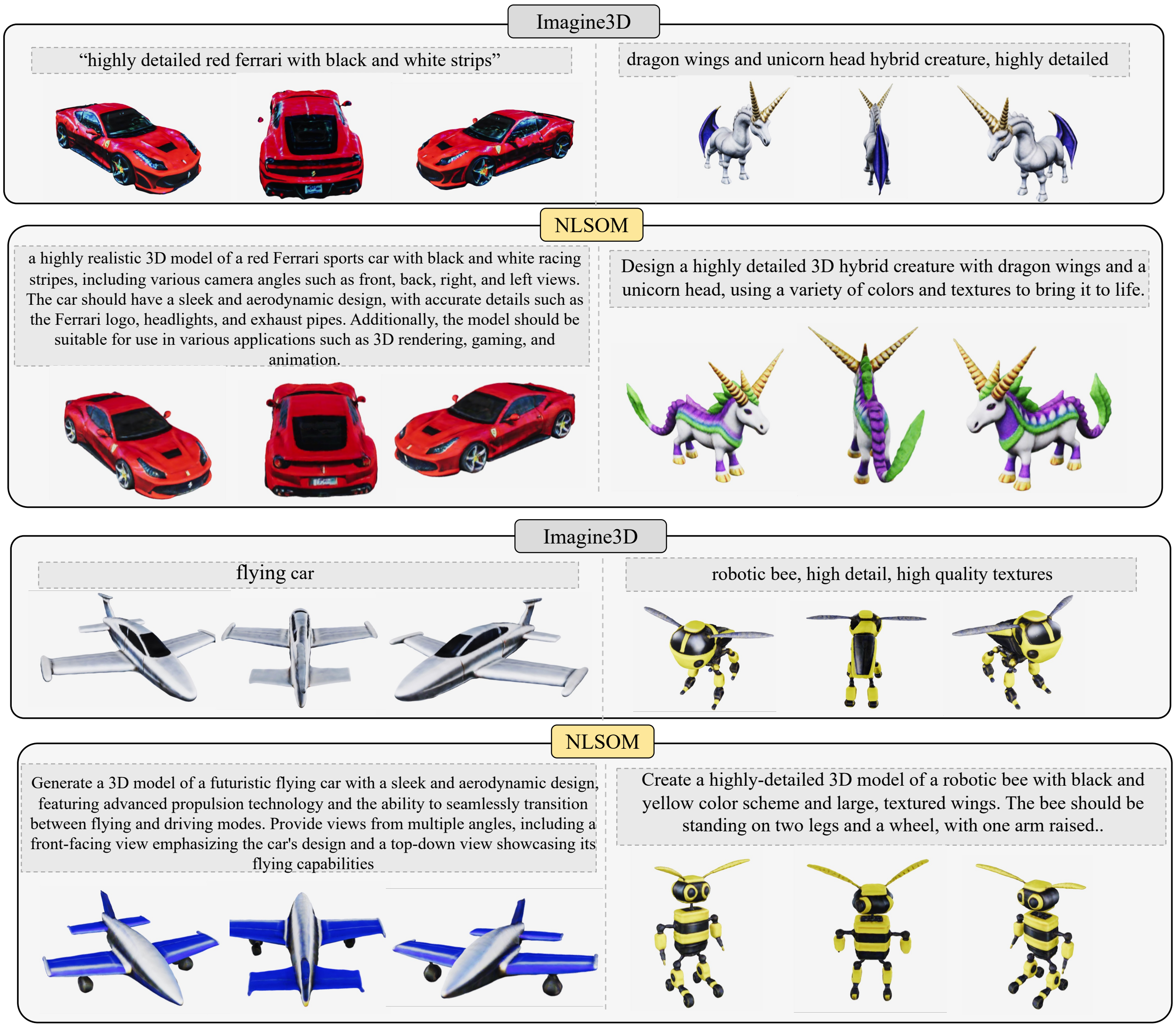


Figure 22: The generated samples from Imagine3D [54] and Imagine3D with our NLSOM. The proposed method enables the injection of open-world knowledge into textual descriptions for 3D model generation. Furthermore, our NLSOM utilizes image caption models to convey information about the initial 3D generation, which improves the visual quality of the final generated 3D content.

Table 5: Quantitative comparisons between our NLSOM and SoTA Imagine3D [54] using average Clip score [56] on different 3D generation tasks.

	3D Generation Tasks Clip Score *100 (↓)				
Model	Ferrari	Unicorn	Flying	Bee	Average
Imagine3D [54]	30.0	**30.0**	28.6	32.3	30.3
+**NLSOM**	**29.0**	**30.0**	**26.0**	**31.3**	**29.1**

APPENDIX H
MORE DETAILS OF EGOCENTRIC RETRIEVAL EXPERIMENTS

H.1 Framework

Objective

Retrieval of a specific scene from memory is a common task in the real world, both in artificial systems, as well as in humans. Developing an intelligent machine capable of solving this problem typically necessitates a sophisticated retrieval system. This system usually incorporates a dedicated model for action/object recognition [237]–[240], multimodal modeling [241]–[251], and temporal action localization [252]–[255]. Our objective is to address the query-from-memory problem from the NLSOM perspective through the modulation and interaction of a diverse set of simulated agents.

The set of agents

We design two types of agents to proceed with the query task. Each debater agent processes a small subset of the inputs and produces a proposition that is relevant to the question. The editor agent compares and summarizes the propositions from several agents, and finally executes the retrieval operation. See below for the characteristics of each type. Type I (Debater agent).

> ***Ability***: *question-relevant information extraction.*
> ***Input***: *natural language.*
> ***Output***: *natural language that describes partial solution candidates to the retrieval problem.*

Type II (Editor agent).

> ***Ability***: *summarization and information integration.*
> ***Input***: *natural language.*
> ***Output***: *natural language that provides a solution to the retrieval problem.*

The protocol We use the following four-step process in our NLSOM: 1) During Mission Initialization,we generate video narrations to provide a detailed description of the egocentric video recording. The description is later given to several Type-I agents, but each agent receives only a partial observation of it; 2) During the Task-Oriented Mindstorm, our agents in NLSOM exchange information about the task. 3) During Opinion Gathering, the information collected during the mindstorm is summarized and analyzed; 4) During the Solution Generation, the models are asked to generate a final output.

(1) Mission Initialization This is an initial stage or pre-processing stage to set up our models for the natural language query task. For the narrations, we utilize high-quality data annotated by humans, enabling us to prioritize the study of interactions between artificial agents. Note that human narration may still introduce subjectivity and biases. After Mission Initialization, we have sufficient textual information to complete the task. A set of Type I agents receive different subsets of the narrations. In the beginning, most of the agents are not confident with their prediction, especially when they have not observed the query-relevant part of the video narration.

Each agent uses the following prompt for providing an initial prediction for the task:

> You are going to answer some questions about a video. Here is a summary of the video: \n <video_summary> \n Followings are the video content. \n <sampled_narrations> \n The video ends here. \n My question is, <language_query>

(2) Task-Oriented Mindstorm

In this phase, we prompt the agents to have rounds of discussions to gather opinions. Each agent receives the latest statement from others and adjusts their output based on their additional guess and the additional information from the other agents.

The following prompt is used during the mindstorm:

Thanks for your answer. Regarding the question, <language_query> I also asked your colleagues <agent_a_name>, <agent_b_name>, and <agent_c_name>. They are all my assistants. They have the observation of the other part of the video. You can choose to trust them or not. You also have your unique observation of the video. Here is what <agent_a_name> says. <agent_a_initialization>. ... What do you think? How much do you agree or disagree with them?

(3) Opinion Gathering All agents are instructed to make their outputs more concise. They are expected to give a clear reason for each predicted video timestamp. We design two social structures to merge agents' summaries from the task-oriented mindstorm. In the monarchy structure, we assign an editor agent and prompt it to produce a final list of possible timestamps. In the democracy structure, we disable the editor agent and ask each agent to vote for the timestamps, and the final list consists of the timestamps sorted by the number of votes. Both structures take the summaries as input and output a candidate list.

We use the following prompt for gathering a final opinion:

I have collected the answers from my assistants to my question, <language_query>. The answers are as follows. <summaries> \n Please carefully read and analyze their answers, then conclude and summarize all the possible answers and the reasons why they are possible answers in a few sentences.

In the democratic structure, all agents utilize the following prompt to gather their votes for the proposed frames:

Now, considering all the conditions, please summarize your final answer to my question. The question is <language_query>.

(4) Solution Generation

The solution generation process optionally post-processes the predictions. We remove invalid predictions, such as negative values. If the gathered opinions lacks diversity (e.g., all predicted frames are less than k), we augment the predictions by appending predictions distributed according to the *grid* baseline (see Section H.2). We also remove duplicated predictions or average the predictions if they are very close to each other, *i.e.*less than one second.

H.2 Implementation Details

Setup: We share a similar working pipeline to NLSOM-VQA as introduced in Section D.2, with minimal adaptation. Specifically, the input to our NLSOM is the text representation of the ego-centric videos, and the natural language query, while the output is an ordered list with predicted video timestamps that match the query. All our agents are based on ChatGPT (`GPT3.5-turbo`).

Dataset: Our experiments are conducted on the Ego4D dataset [59], which is a large egocentric dataset for daily activity recordings of humans. We validate our algorithm on the 5% of the *val* split of the Natural Language Query (NLQ) task. This results in 192 unique query pairs from ~100 videos; each query pairs consist of a video clip and a query expressed in natural language. The target is to localize the temporal window span within the video history where the answer to the question can be found.

Metrics: To directly compare models, we use the top-k recall (R_k) as our evaluation metric. Given a predicted ranked list from a query, we compare the first k predictions $t_1, \cdots, t_k$ with the ground truth temporal span, denoted as t_s, t_e. If any of the predictions are in between t_s, t_e, $\exists t_i \in \{t_1, \cdots, t_k\}, t_s \le t_i \le t_e$, we count this prediction as positive. Moreover, since the visual information is missing from the model, we relax the condition by a threshold τ, *e.g.*, $tau = 10s$. The condition of relaxed top-k recall becomes $\exists t_i \in \{t_1, \cdots, t_k\}, t_s - \tau \le t_i \le t_e + \tau$, denoted as $R_k@\tau$. Empirically, we have $k = 1, 3, 5$ and $\tau = 1, 10$ seconds.

Baselines: We compare our results with two heuristic baselines, *random* and *grid*. In our random baseline, we randomly pick a timestamp between the beginning and the end of the video sampling from a uniform distribution. This is repeated k times to compute R_k. For the grid baseline, we evenly divide the video into $k+1$ components and take the boundary frame between components as the predictions. Moreover, we built another baseline, denoted as *individual*, showing the performance when there is only single agent to localize the query.

Supervised methods: Besides our line of work, recent supervised learning methods also show promising results on the NLQ task [256]–[259]. We compare our method also with DenoiseLoc [260], which is a state-of-the-art supervised method that uses video frames as inputs. The performance of NLSOM, which employs the collaboration of a set of zero-shot learners, is relatively lower due to a lack of dataset/task prior. However, state-of-the-art supervised methods can in principle be augmented with ours proposed techniques for coarse-to-fine localization [261]. We leave this for future work.

H.3 Performance Analysis

Analysis

We present our result in Table 6. In our first setting, we use a single agent (denoted as *individual* to solve the retrieval task and compare it to our *random* and *grid* baselines. The *individual* agent directly observes the video frames. We do not observe a strong improvement of a single LLM over our baselines *random* and *grid*. Although the accuracy metrics, R1 and R1@1s, are higher, they are still close to random selection. This demonstrates the complexity of the memory retrieval task: it is difficult for a single agent.

The Social Structure of this NLSOM

The third and fourth row of Table 6 show model performance with larger numbers of agents, implemented in two distinct social structures. Both of them consistently surpass the heuristic baselines and single-agent experiments. We hypothesize that this is because each agent in the society reasons better when it is tasked with a simpler objective. Furthermore, since the data input of each agent is different, they are able to generate unique insights to the problem, resulting in a more diverse set of predictions. We also compare the two social structures, monarchy, and democracy. In our setting, we observe that a democratic structure works slightly better. One reason for this may be that the agents vote directly based on their observation and the ideas from other agents, while the monarchical structure requires an additional conclusion round where critical information has a higher chance of being corrupted. Nevertheless, we also observe that a monarchy can work better with multiple model prediction trials, evident by a higher top-5 recall. When applying multiple rounds of discussion, the retrieval performance is not significantly improved. This may either be due to efficient communication between agents in a single round or loss of information when multiple rounds are applied.

Importance of the Solution Generation Phase

We also experiment with a modified *solution generation* phase where a common post-processing script is applied, denoted as *w. exec*. Specifically, we ensure the number of predictions is greater or equal to k, and all of them are in a valid range, such as $[0, T]$, where T is the timestamp of the last video frame. This leads to improvements as it can be seen in Table 6.

Visualization

We first show a typical example from the dataset in Figure 23. Here, the scene to retrieve is the activity of removing an object from a car. In the mindstorm session, only the first agent is able to localize the target activity, and all the others do not find any indication for it, because the recorder went to a room, and the car is out of view. However, after a few rounds of discussion, the other agents agree that the target activity is before or after the video they processed and trust the message from the first agent. Finally, a Type-II agent that didn't observe any activities itself, makes a final prediction which successfully includes the target frame. We believe that with the help of visual localization experts, such as G-TAD [262] or VLG-net [243], the performance of such retrieval system can be further improved.

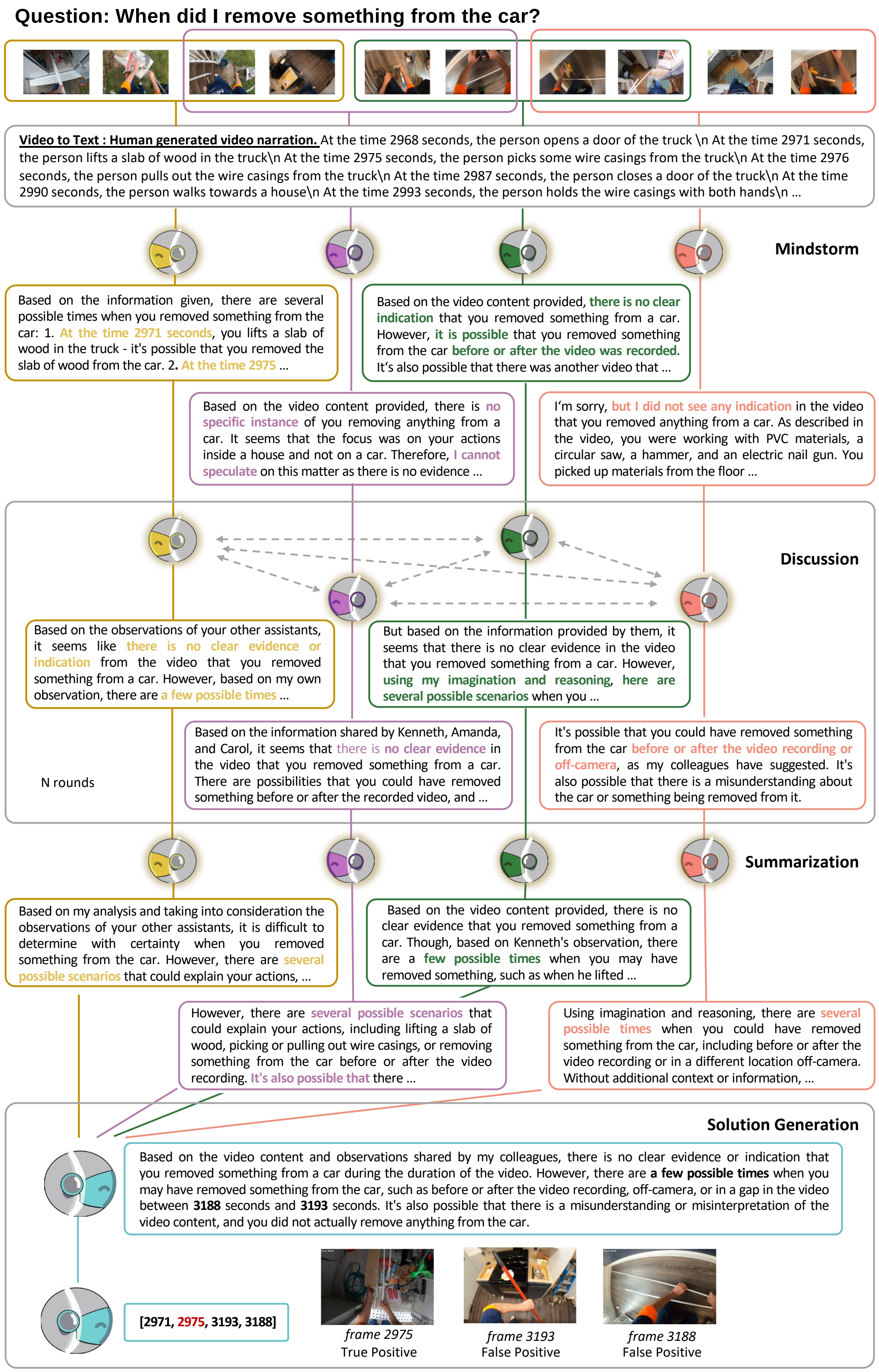


Figure 23: An example from Ego4d. We ask our model to localize the activity of removing something from the car. Only the first agent is able to see the target activity at the beginning of the video. The other agents obtain relevant information from the first agent after several rounds of discussions. The final prediction successfully recovers the ground truth.

Table 6: Our benchmark for egocentric retrieval with NLSOMs. We compare NLSOMs to random and supervised learning baselines. All recalls are computed on a 5% fraction of the NLQ validation set of the Ego4D dataset. *Random* and *grid* are heuristic baselines, *supervised* is DenoiseLoc [260], and *w. exec.* describes an additional post-processing step after opinion gathering. Note that all the experiments are zero-shot learning settings except for the last row being supervised learning. *m.* for monarchy, and *d.* for democracy.

method	# of agents	Rounds	*R1*			*R3*			*R5*		
			-	*@1s*	*@10s*	-	*@1s*	*@10s*	-	*@1s*	*@10s*
random	0	N/A	4.69	5.21	12.50	7.29	7.81	20.31	10.42	11.46	30.21
grid	0	N/A	3.12	3.65	7.81	6.25	7.81	15.10	11.98	14.06	27.60
individual	1	N/A	5.21	6.77	9.38	9.38	10.94	15.10	10.42	11.98	16.15
Ego-NLSOM (*m.*)	4	1	6.77	8.33	9.90	14.58	16.15	21.88	**19.27**	**20.83**	25.52
Ego-NLSOM (*d.*)	4	1	**8.85**	**9.35**	**14.58**	**16.67**	**16.67**	**23.96**	18.75	20.31	**28.12**
Ego-NLSOM (*m.*) *w. exec.*	4	1	7.81	9.90	13.02	16.67	**18.75**	**29.17**	**25.00**	**27.08**	**40.62**
Ego-NLSOM (*d.*) *w. exec.*	4	1	**9.38**	**10.42**	**17.19**	**17.71**	18.23	**29.17**	22.92	25.00	39.58
Ego-NLSOM (*m.*) *w. exec.*	4	2	7.14	8.79	13.19	14.29	17.03	24.18	23.63	25.82	37.36
Ego-NLSOM (*d.*) *w. exec.*	4	2	8.85	**11.98**	**16.67**	**16.15**	**18.75**	**29.18**	**23.44**	**27.05**	**40.10**
DenoiseLoc [260]	N/A	N/A	20.01	25.76	34.77	24.99	30.82	39.47	27.85	32.94	42.13

APPENDIX I
MORE DETAILS ON THE EMBODIED AI EXPERIMENTS

We explore how to leverage pre-trained LLMs in two different settings: autonomous exploration [263] [264], which is recognized to be a fundamental problem in the field [265], and Episodic Question Answering (EQA) [266]. NLSOMs enable robots to complete these tasks in a zero-shot fashion.

I.1 The set of agents

In order to solve these tasks, we divide the problem into three subtasks: (1) Produce natural language descriptions from a sequence of egocentric videos obtained through a robot's sensors (2) Reason about the action that the robot needs to take, using the previously generated descriptions. (3) Answer questions about the explored environment.

We design an NLSOM system comprising three agents to tackle these subtasks as follows:

Type I (Observer).

> ***Ability***: *given a question, describe visual data in natural language.*
> ***Input***: *language-based questions and visual data* i.e., *RGB videos, RGB-D videos.*
> ***Output***: *natural language describing the visual data.*

Type II (First Mate).

> ***Ability***: *summarize and reason.*
> ***Input***: *natural language.*
> ***Output***: *contextual questions in natural language based on previous questions and answers; and a summary of gathered information.*

Type III (Captain).

> ***Ability***: *summarize and reason.*
> ***Input***: *natural language.*
> ***Output***: *provide a description in natural language outlining the specific actions required by the robot, while also offering responses to questions based on the observed surroundings.*

I.2 The protocol

- *Mission Initialization.* All agents are initialized with their respective prompts given in Table 7.
- *Task-Oriented Mindstorm.* The generated captions from VLMs often lack intricate details and appear rough when generated based on individual questions. Additionally, VLMs face additional difficulties due to the low-quality observation frames obtained from the environment simulator, as shown in Figure 24. We introduce a task-oriented mindstorming procedure to address this challenge.
  In order to generate rich and accurate language-based descriptions for visual environments observed by a robot, an agent of Type I and of Type II, each with different abilities, collaborate in natural language. More specifically, the Type II agent generates questions related to the environment. With these questions, the Type I agent (*i.e.*VLM) can describe different aspects or regions of a video frame, instead of expressing the whole content of the frame at a time. Furthermore, information is aggregated across multiple frames, which may further improve predictions even with low-quality frames. In our experiments, there are a total of 10 rounds of questioning and answering, where each new question is conditioned on the entire previous conversation.
- *Opinion Gathering.* The Type II agent summarizes the results of the mindstorm procedure in order to provide Type III with concise descriptions of the environment. We assume that the capabilities of the Type III agent include real-world knowledge as well as language understanding and abstraction. The Type III agent examines and summarizes the information from an environment that is only partially observable. It utilizes its inherent real-world knowledge to determine the most appropriate action to be taken next. It has access to the entire interaction history of previous observation summaries and taken actions.
- *Execution.* Given a question or action request, the Type III agent generates answers. If the task is exploration, the Type III agent produces an action that is taken by the virtual robot in the next step.

I.3 Implementation Details

Setup. We adopt BLIP2 as the Type I agent, and both Type II and III agents are based on ChatGPT.

Simulated robot and environment. We use the Habitat [60] simulator based on the Matterport 3D dataset (MP3D) [61] which contains various indoor scenes. In our study, we utilize the established division of the MP3D dataset based on the PointNav task. Additionally, we specifically select single-floor houses to facilitate the evaluation of embodied exploration experiments. The habitat simulator enables the robot to move in the virtual environment. The action space is identical to the PointNav [267]. The available actions are: move forward, turn left, turn right, and stop. The action *move forward* directs the robot to move forward for a fixed distance of 0.5 meters. Meanwhile, *turn right* and *turn left* instruct the robot to rotate 45 degrees to the right or left, respectively. To achieve human-like observations for the Observer agent, we equip the robot with an RGB camera positioned at a height of 1.5 meters.

Prompts. The prompts for each agent are given in Table 7.

I.4 Performance Analysis

We conduct a qualitative evaluation of our method's performance on zero-shot EQA. Figure 24 illustrates that our method successfully provides accurate answers to challenging questions. Notably, our method demonstrates the capability to recognize and understand the spatial relationships among objects within the 3D environment. Additionally, the figure highlights that our method can reasonably answer open-ended questions such as *Is this house good for living*.

Table 7: The prompts used for each agent type in the embodied AI experiments.

Type	Instructions
Type I. Observer (to First Mate)	I've snapped a photo of a room in the house. Please ask me detailed questions about the image content, specifically the objects and their relative positions within the room. As I'll be relying on your textual narration to navigate later, it's crucial to ask informative questions to maximize my understanding of the room.
Type II. First Mate (to Observer):	Please respond to the provided questions truthfully. If you are unsure of an answer, simply state that you do not know. It is important not to describe any content that is not present in the image.
Type III. Captain (to First Mate):	Use several sentences to summarize the information you get about this indoor scene.
Type II. First Mate (to Captain):	You {action}, in this place, you have seen {observation}.
Human (to Captain):	Please control the robot to fully explore this house. At each step, your first mate will update you with the new observations caused by the most recent action. Please tell the robot the next action based on previous actions and observations.

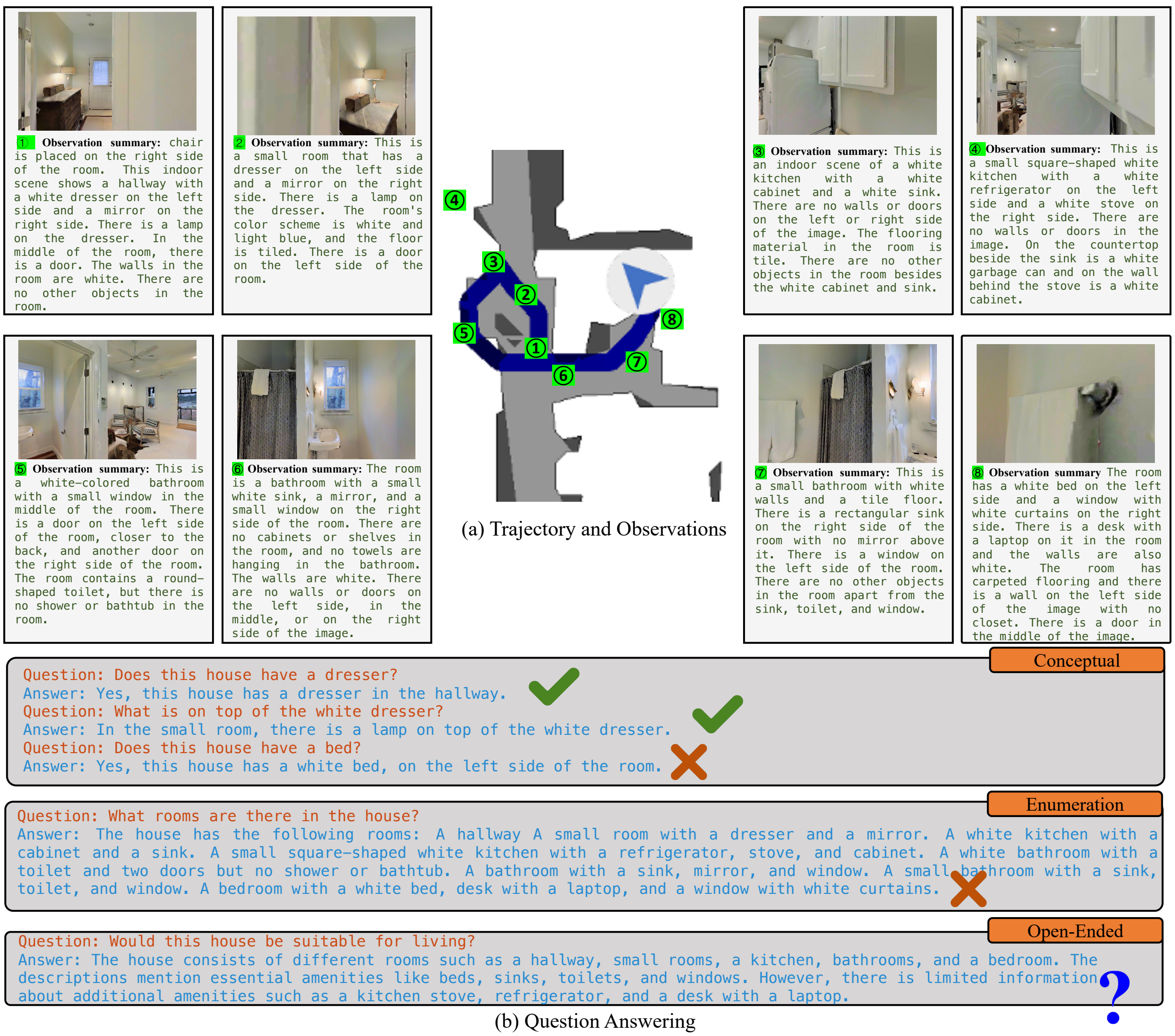


Figure 24: Qualitative results of embodied question answering.

APPENDIX J
MORE DETAILS OF GENERAL LANGUAGE-BASED TASK SOLVING

In this section we describe our framework, which leverages the power of collaboration among multiple chat agents to successfully complete assigned tasks. Each agent is carefully assigned a role that aligns with the needed skill set and area of expertise for completing the task. The roles could be either assigned by a human or another agent. The agents are then set to work together in a cooperative and coordinated manner to accomplish a specific goal.

J.1 The set of agents

In our framework, all individual agents possess a common input, ability, and output. This design mirrors the interactive nature of human societies, in which communication occurs primarily through the use of natural language. Specifically, the agents' shared ability is to comprehend and analyze natural language, while their input and output channels are also based on this mode of communication. To summarize: the problem-solving process within our multi-agent system is founded on the agents' ability to process and interpret natural language.

Ability: *understand and analyze the input natural language presented to it.*
Input: *natural language input which reflects a task, instruction, question or any other informative text.*
Output: *natural language output which reflects a reply to the presented input.*

J.1.1 Setup & Protocol

CAMEL [63] is a novel role-playing based framework to achieve a scalable approach that facilitates autonomous cooperation between communicative agents. We adopt this role-playing framework and use the same "inception prompts" to assign different social roles to multiple `GPT3.5-turbo` agents. The agents are then asked to communicate collaboratively to solve an assigned task of interest representing a realistic AI society setting. Through this framework, we could explore the "mind" of these agents and understand their cooperation capabilities, behavior, and failure cases.

As stated earlier, CAMEL is designed to automate problem-solving tasks through cooperative communication between multiple agents. When a human requires assistance with a task, CAMEL follows the following process:

1) **Role Assignment**: The human assigns two agents roles that are appropriate for solving the task based on their skill sets and expertise.
2) **Task Specification** *(Optional)*: If necessary, an agent can be utilized to help the individual refine and enhance the task.
3) **Role-Playing**: The two agents work together using natural language to solve the main task, with the assistant agent completing tasks assigned by the user agent to ultimately reach a final solution for the task.

By following this process, CAMEL streamlines problem-solving tasks, utilizing the expertise of multiple agents to deliver a solution to the assigned task.

J.1.2 Experiments

Two Agent Society (Cooperating for Good Example). We begin by presenting a minimalistic society composed of only two agents (no task specifier). The task of interest to the human leader in this example is the to *"Create a simple dice flip game with a cool user interface using Python. If the user flips an odd number they lose otherwise they win."*

As mentioned in Section J.1.1, if no task specifier is used, two steps are to be carried out:

- *Role Assignment:* Each agent is assigned a particular role to fulfill in the society and is asked to cooperate with the other agent to solve a task of interest. For solving this task, we assign one agent the role of an assistant programmer and the other agent as a game developer. Those skill sets are aligned with the task we are solving.
- *Role-Playing:* The two agents have several rounds of mindstorm and are capable of building up the game without any sort of human involvement. The UI is shown in Figure 26. The output code generated by the assistant is shown in Figure 25.

Dice Game Generated with Task Specifier. Including a third agent in the development of the dice game, namely the task-specifier agent, leads to a more interesting and sophisticated game compared to the two-agent setting.

As described in Section J.1.1, three steps are carried out:

- *Task Specification.* The human leader first gives the instruction, "Create a simple dice game using Python." Then, a *task specifier* provides the idea: "Include a GUI with a welcoming image, allow the player to input their name, have the

```python
import tkinter as tk
import random

root = tk.Tk()
root.title("Dice Game")
root.geometry("400x400")

canvas = tk.Canvas(root, width=200, height=200)
canvas.pack(pady=20)

def draw_dice(number):
    canvas.delete("all")

    if number == 1:
        canvas.create_oval(75, 75, 125, 125, fill="black")
    elif number == 2:
        canvas.create_oval(25, 25, 75, 75, fill="black")
        canvas.create_oval(125, 125, 175, 175, fill="black")
    elif number == 3:
        canvas.create_oval(25, 25, 75, 75, fill="black")
        canvas.create_oval(75, 75, 125, 125, fill="black")
        canvas.create_oval(125, 125, 175, 175, fill="black")
    elif number == 4:
        canvas.create_oval(25, 25, 75, 75, fill="black")
        canvas.create_oval(125, 25, 175, 75, fill="black")
        canvas.create_oval(25, 125, 75, 175, fill="black")
        canvas.create_oval(125, 125, 175, 175, fill="black")
    elif number == 5:
        canvas.create_oval(25, 25, 75, 75, fill="black")
        canvas.create_oval(125, 25, 175, 75, fill="black")
        canvas.create_oval(75, 75, 125, 125, fill="black")
        canvas.create_oval(125, 125, 175, 175, fill="black")
        canvas.create_oval(25, 125, 75, 175, fill="black")
    elif number == 6:
        canvas.create_oval(25, 25, 75, 75, fill="black")
        canvas.create_oval(125, 25, 175, 75, fill="black")
        canvas.create_oval(25, 75, 75, 125, fill="black")
        canvas.create_oval(125, 75, 175, 125, fill="black")
        canvas.create_oval(25, 125, 75, 175, fill="black")
        canvas.create_oval(125, 125, 175, 175, fill="black")

def roll_dice():
    number = random.randint(1, 6)
    draw_dice(number)
    if number %
        message_label.config(text="You win!")
    else:
        message_label.config(text="You lose!")

roll_button = tk.Button(root, text="Roll Dice", command=roll_dice)
roll_button.pack()

message_label = tk.Label(root, text="")
message_label.pack(pady=10)

root.mainloop()
```

Figure 25: Dice Game Cooperative Solution. The code for the dice game generated after multiple rounds of discussion between the two agents is shown here.

game roll two dice, and display a message with the sum and whether the player won or lost. The player wins if they score a sum greater than 6."

- *Role Assignment.* In this step, the role assignment is based on needed expertise, so *CAMEL* assigns an agent 1 (*Python Programmer*) and an agent 2 (*Game Developer*).
- *Role-Playing.* The two agents have several rounds of mindstorm. In the first round, the *Python Programmer* says, "Import the necessary libraries for creating a GUI," and the *Game Developer* replies, "To create a GUI, we need to import the tkinter module in Python. Here's an example code: `import tkinter as tk`." In the second round, the *Python Programmer* says, "Create a tkinter window with a title and a fixed size," and the *Game Developer* replies, "I will use the `Tk()` method to create the window object, and then use the `title()` and `geometry()` methods to set the title and size, respectively. Here's an example code: `import tkinter as tk` `window = tk.Tk()` `window.title("Dice Game")` `window.geometry("400x300")`". The mindstorm ends when the instantiation of ChatGPT is out of tokens, when it exceeds a threshold number of messages (50 in our case), or if the task is "done" according to the user agent.

Interestingly the agents are able to create a game where the user wins if the sum of the two rolled dice is greater than 6 and loses otherwise. The solution of the task also requires adding an image as the specified task prompt requires having a "welcoming image". We generated this image using stable diffusion. The new game GUI and sample runs of different users is shown in Figure 27.

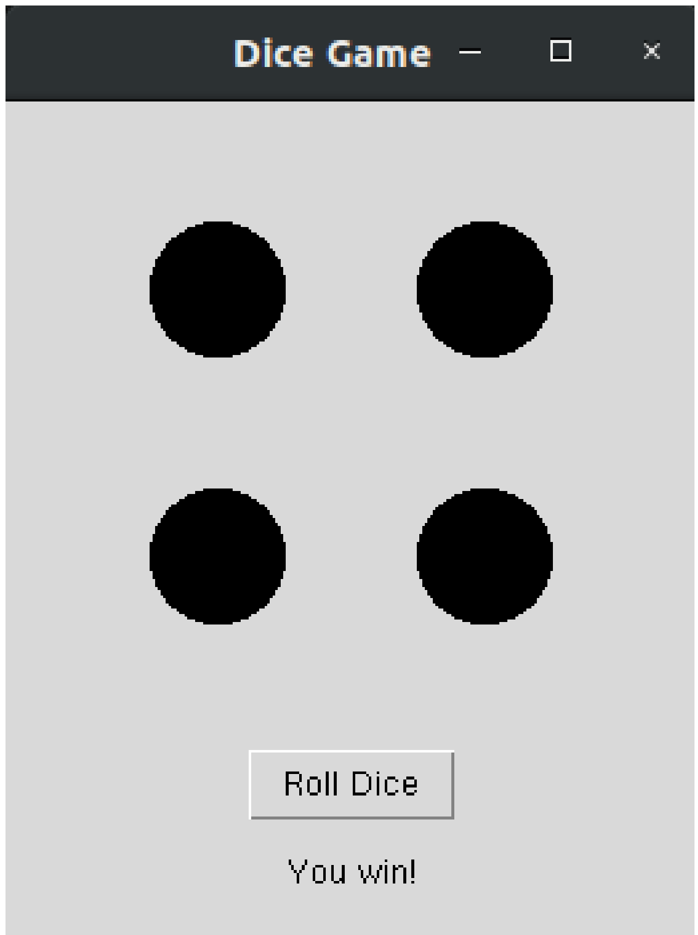


Figure 26: CAMEL Simple Dice Game. Two agents are capable of creating a dice game that works directly out of the box without any human interference.

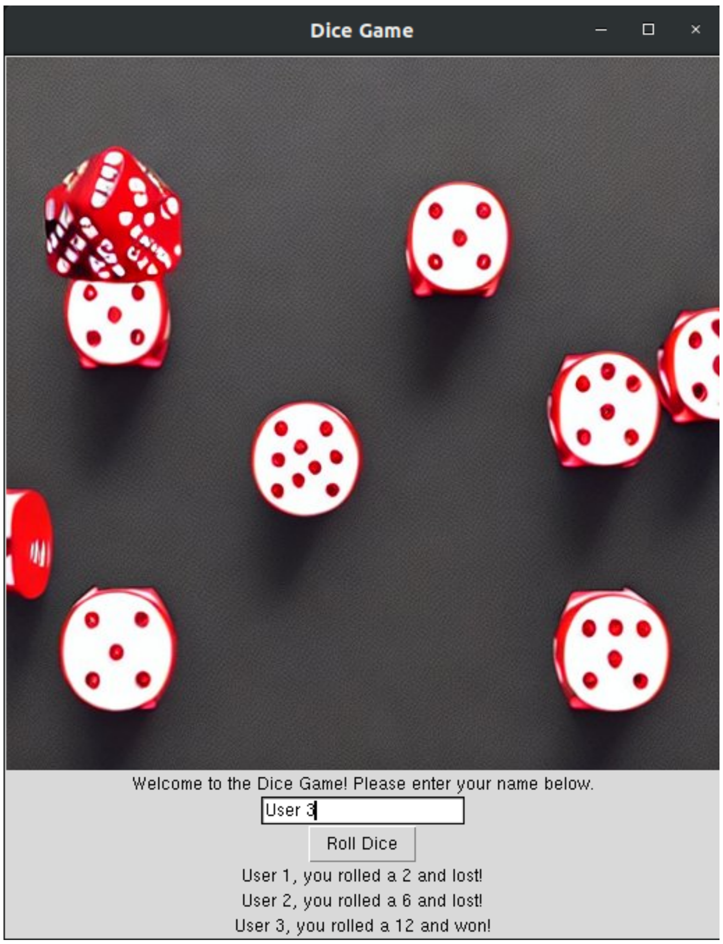


Figure 27: Dice Game Generated with Task Specifier.