A computational framework for learning and transforming task representations

Andrew Kyle Lampinen

Humans exhibit substantial cognitive flexibility. We can use our knowledge of a task to adapt to novel variations on our first try. For example, imagine you are playing poker with your friends, and one of them says “next round, let’s try to lose.” You will be able to perform well, despite the new goal directly contradicting your prior goal. Alternatively, suppose your friend says “next round, threes will be wild.” You will be able to play this variation, even if you have never played it before. For an example from another domain, imagine your friend shows you a blue car, and then tells you that their car looks similar, except that it is red. You will be able to combine these pieces of knowledge to recognize their car, despite never having seen it before. Humans can often adapt our knowledge to a new situation on our first try.

By contrast, while deep learning models can achieve human (or super-human) performance at games (Silver et al., 2017; Vinyals et al., 2019) and object recognition (Szegedy et al., 2016), they are unable to flexibly adapt their knowledge of these tasks (Lake et al., 2017). How could a deep learning model trained to win at poker reuse that knowledge to try to lose? How could a model trained to recognize one car adapt to recognize that car in a different color? Many deep learning models cannot, even in principle.

Observations like these have formed a key piece of contemporary critiques of deep learning models as cognitive models (Lake et al., 2017; Marcus, 2018). These critiques echo criticisms of the generalization capabilities of neural networks from the earlier days of connectionism (Fodor and Pylyshyn, 1988). Can deep learning serve as a foundation for plausible cognitive models, or is the approach fundamentally limited, as these critiques suggest?

In this dissertation, I bring new light to this debate, by proposing and demonstrating a novel approach to adaptation in deep learning models. This approach relies on the idea that task relationships are key to generalization—indeed, it is only insofar as their are systematic relationships between tasks that generalizing to new tasks could be possible. Thus, I propose to use task relationships explicitly when performing a new task.

In particular, I propose to interpret task relationships as transformations of tasks. That is, we can model a relationship between tasks A and B as a higher-order function which takes task A as input and outputs task B. For example, there might be a "try-to-lose" transformation that would take poker as an input, and transform it to a losing variation of poker; that would transform chess to a losing variation of chess; etc. I therefore propose a general framework for both learning task representations, and learning to transform those task representations in order to adapt to new tasks. This learning-based approach does not require building in prior knowledge about the structure of the tasks, or compositional symbolic reasoning. Instead, the transformations are learned solely from relationships between learned representations of tasks.

My approach allows deep learning models to flexibly reuse their knowledge of a task to adapt to novel variations of that task. In fact, my approach demonstrates 80-90% performance on novel tasks across a broad range of domains, including mathematical objects, simple card games and video games, and visual classification. It achieves this performance even in challenging settings where the new tasks directly contradict prior knowledge.

Furthermore, this adaptation is not just useful on the first time performing a task—it subsequently allows the model to continue learning and perfect its performance on the task much more efficiently than alternative approaches. This shows that deep learning models can flexibly reuse their knowledge to perform well on their first try, and learn rapidly from this starting point, without requiring built-in prior domain knowledge, or explicitly symbolic representations and reasoning systems.

This work therefore has broad implications. It addresses fundamental questions about the computational approaches necessary for flexible intelligence, and how that flexibility can contribute to later learning. This has implications for cognitive science, neuroscience, and philosophy of mind. In addition, I demonstrate the approach within several important contemporary artificial intelligence paradigms, including visual classification and deep reinforcement learning. It may therefore also be of interest to researchers who wish to build more flexible artificial intelligence and machine learning systems.

1 The method: adaptation as task transformation

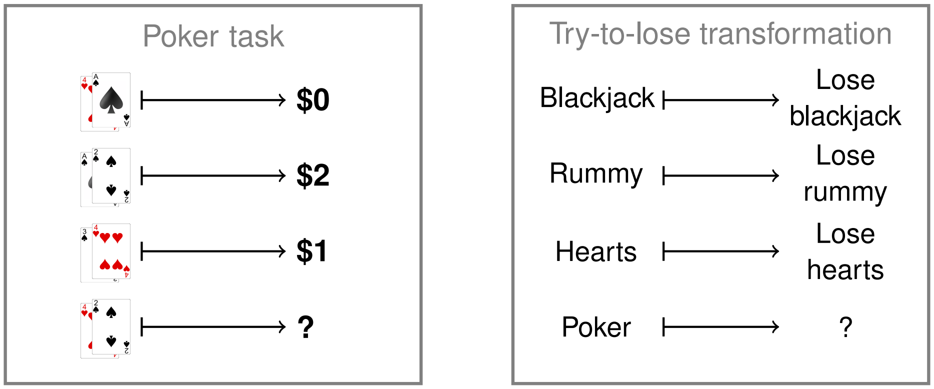
My approach begins with the idea that tasks can be seen as mappings from inputs to outputs. For example, poker can be seen as a mapping of hands to bets (Fig. 1a), and visual classification can be seen as a mapping from images to classifications. I suggest that to perform many of these tasks flexibly, as humans can, we must be able to constrain these mappings by an internal representation of the current task. I therefore draw inspiration from both cognitive science and machine learning, and allow my model to construct a task representation from either a language cue, or examples of the task (Fig. 2a).

The model then uses its internal representation of the current task to adapt to that task (Fig. 2b). Specifically, the task representation adapts the weights of a task network so that it can make appropriate decisions in the task, for example by betting highly on straight flushes when playing poker.

The key insight of my dissertation is that the task representations that the model has constructed can then be transformed to adapt to new tasks. By transforming a prior task representation, the model can perform a new task without having experienced it, just as humans can adapt to new tasks by transforming their prior knowledge. Specifically, I allow the model to construct a representation of a task transformation from a language cue (“try to lose”) or examples of the transformation (trying to win and lose at various other games), see Fig. 2c.

This representation of the transformation can then be used to adapt to that transformation, by setting the weights of a task network to appropriately transform task representations (Fig. 2d). For example, the system could apply the “try-to-lose” transformation to its representation of poker to produce a representation of the task of losing at poker. This representation could then be used to perform that task (Fig 2d-detail).

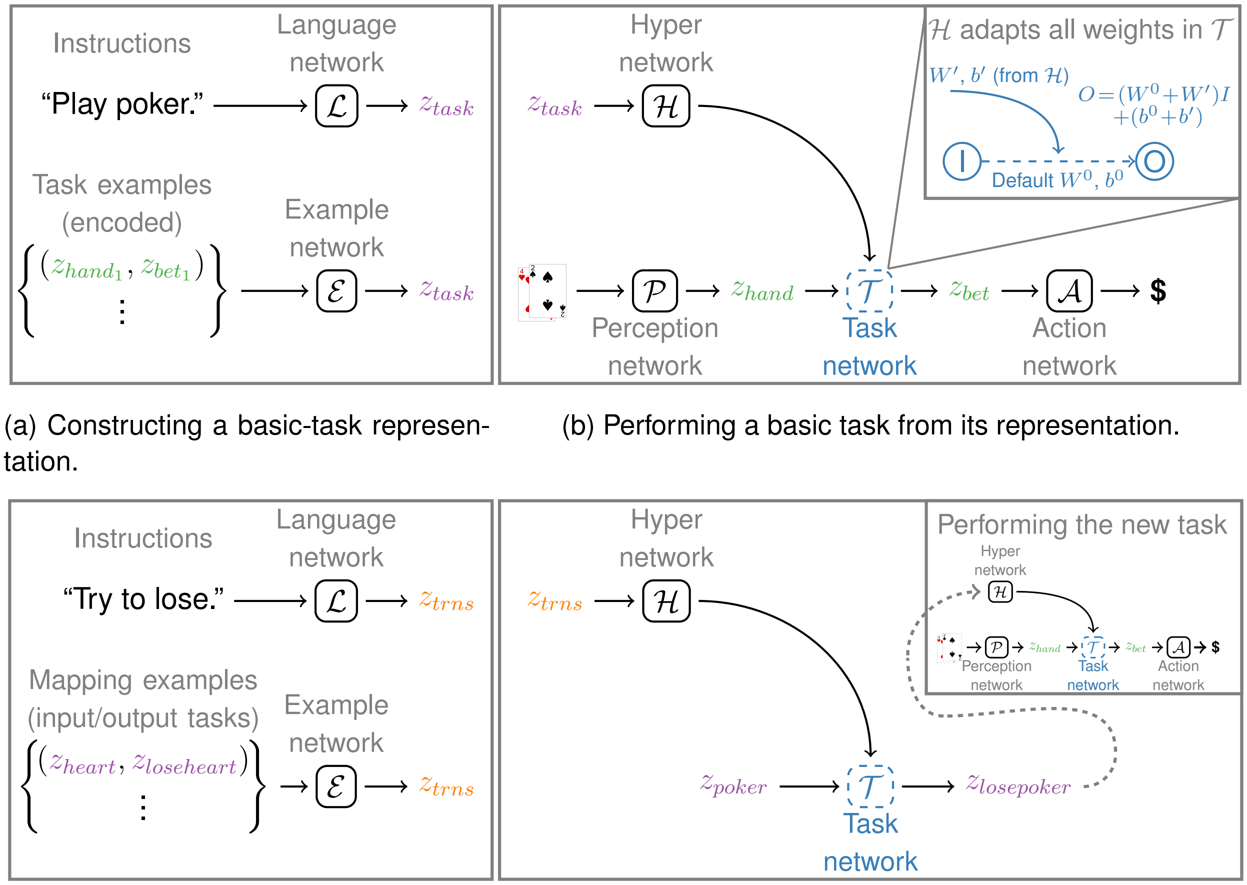
There is an analogy between the basic tasks and meta-mappings—both are simply functions from inputs to outputs, just of different types. The implementation I propose, called the HoMM architecture, therefore parsimoniously reuses the same architectural components for both basic tasks and transformations of basic tasks. This parallel is reflected in the parallels between the top and bottom rows of Fig. 2. This approach is parsimonious, in that it does not multiply networks unnecessarily. It also reflects ideas from the theory of programming languages, and aspects of the Global Workspace Theory of consciousness (Baars, 2005). (These connections will be briefly discussed below.)



(a) A basic task.

(b) A task transformation.

Figure 1: Basic tasks and task transformations. (a) Basic tasks can be seen as mappings from inputs to outputs, for example, from poker hands to bets. Tasks can be generalized from examples. (b) Task transformations are higher-order tasks, which take a basic task as input, and output a transformed version of that task, for example, switching from winning to losing a game. Task transformations can be generalized from examples.



(a) Constructing a basic-task representation.

(b) Performing a basic task from its representation.

(c) Constructing a task-transformation representation.

(d) Transforming a basic task representation, by using a representation of the task transformation.

Figure 2: An architecture that performs and transforms tasks. (a,c) The HoMM architecture performs basic tasks and task transformations from a task representation, which can be constructed from appropriate language inputs or examples. (b) The task representation is used to alter the parameters of a task network (see detail) which executes the appropriate task mapping. (d) The task transformation representation is used to parameterize the task network to transform a task representation. The transformed representation can then be used to perform the new task without any direct experience of the new task (see detail). The HoMM architecture exploits a deep analogy between basic tasks and task transformations—both can be seen as mappings of inputs to outputs, although they have different types of inputs and outputs. Thus, the architecture uses type-specific models to embed all basic inputs, as well as tasks and task transformations, in a shared representational space. Then all tasks and task transformations can be seen as transformations applied to entities in this space, which can be executed by shared systems. The parallels between the basic tasks and the task transformations are reflected in the parallels between the top and bottom rows of the figure.

2 Selected experimental results

Card games: I demonstrate my approach in a domain of simple card games. I show that my architecture is able to learn to play card games nearly optimally, and from seeing examples on other games, it is able to transform its representation of simplified poker to play a novel losing variation of that game on its first try, and achieve 85% of optimal performance on this difficult adaptation. This is achieved without giving the system prior knowledge of winning, losing, or any of the concepts relevant to the games, based only on performing tasks and seeing the relationship between winning and losing at other games.

I compare this result to the adaptation of human participants on the same card game. The human participants do not learn to play the game optimally within the limited experiment, but do adapt reasonably well, obtaining nearly equal performance on the winning and losing variations of the game. However, there is substantial variability within and across subjects, which makes it difficult to compare humans to the model—for example, some humans improve substantially on the losing variation of the task compared to the winning variation, even though they receive almost no information about their performance between these phases. However, my model at least achieves higher performance than the humans on the losing variation, and a relative performance change between the winning and losing variations that is not significantly different from the humans. I also show that my method substantially outperforms an alternative approach based on a language description of the new task.

Visual concepts: I also demonstrate my approach in a visual concepts setting (Fig. 3). I show that the system is able to adapt compound concepts, for example adapting from recognizing red triangles to blue triangles, or from recognizing yellow circles to yellow squares. After training on a few hundred visual concepts, the system is able to achieve perfect adaptation to novel concepts based on their relationship to prior concepts.

I instantiated the visual concepts as mappings from raw input images to classifications, as is the standard in computer vision. By providing a mechanism for transforming representations of these tasks, the results of these experiments suggest a method for building vision models a step closer to human-like adaptibility. This could be important both because computer vision is a growing field, and because deep learning vision models are increasingly used as models of human and animal perception (Yamins et al., 2014; Kriegeskorte, 2015),

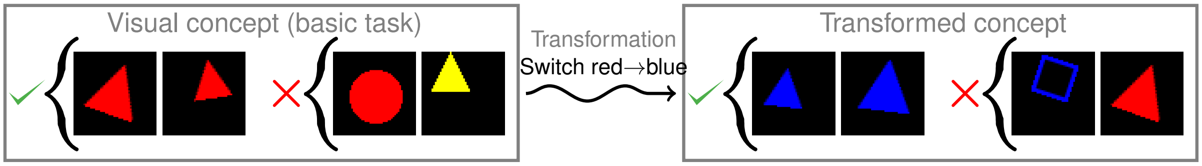


Figure 3: The visual concepts domain. Visual concepts can be thought of as tasks of classifying images as positive or negative examples of the concept. For example, a concept might be red triangles, as shown on the left. These concepts can be transformed by a task transformation that alters some of their attributes, for example switching red to blue, to recognize the concept of blue triangles.

Video games: I also demonstrated my approach in a set of simple video game (reinforcement learning) tasks (Fig. 4). These games required the system to complete a sequence of actions, such as moving around to pick up all the objects of one color in a room, or to push them out of the room, while avoiding negatively rewarding objects of another color. In these games, the system had to learn to adapt to switches of which objects were good or bad, by transforming task representations. The model is able to achieve 80% of optimal performance on average at a novel transformed task, based on examples of the transformation applied to less than twenty tasks. Furthermore, it is even able to generalize from switching colors to switching shapes, even if it has never experienced a transformation of switching shapes.

The deep reinforcement learning paradigm has driven some of the recent artificial intelligence successes at challenging games (Silver et al., 2017; Vinyals et al., 2019). Furthermore, reinforcement learning computations appear to explain some aspects of neural activity (e.g. Dabney et al., 2020). Finally, reinforcement learning requires adaptation of not just an instantaneous decision, but of longer-term actions. Thus, the results of these experiments suggest that my task transformation approach may be broadly useful, and applicable in artificial intelligence and neuroscience research.

Task transformation as a starting point: One key reason why task transformation is use-

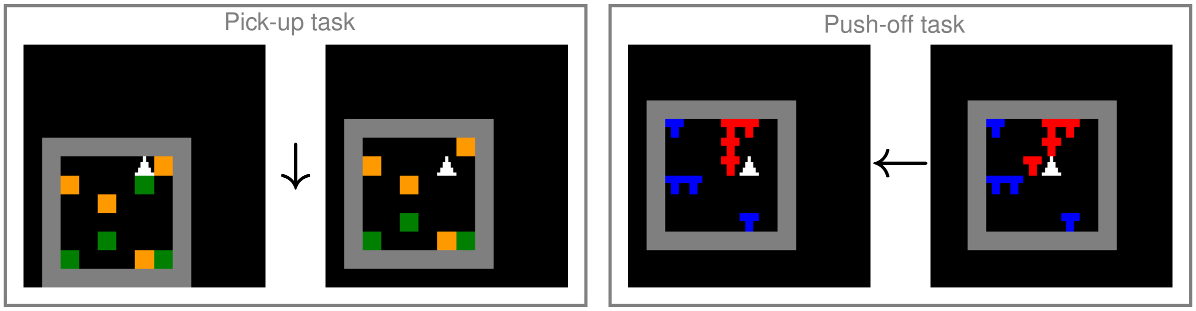


Figure 4: Illustrative state, action, state transitions from the video game experiments. In the pick-up task example (top), the agent moves downward and picks up the green object. In the push-off task example (bottom), the agent moves left and pushes the red object. Views are the visual input the agent would receive. The agent is the white triangle.

ful is that it offers a starting point for later learning. I show that learning from a transformation of a prior tasks allows the system to master a new task while making an order of magnitude fewer mistakes than the next best approach I considered. Ideas like this could be useful for domains like robotics, where mistakes during learning can be dangerous (Turchetta et al., 2016). Furthermore, by optimizing the task representation without altering the model parameters, this learning can be achieved without even the possibility of interference with prior knowledge. This may explain some of the efficiency of human learning.

The analogy between basic tasks and task transformations: I also show that the parsimonious implementation I noted above, which shares architectural components between the basic tasks and transformations of basic tasks, significantly improves performance. This raises interesting questions about the benefits of a shared workspace. For example, the Global Workspace Theory of consciousness (c.f. Baars, 2005) posits that conscious awareness involves bringing information into a shared space where it is accessible to many processes and readout systems. My results highlight that a shared space may have benefits to generalization that complement its benefits to flexible readout and computation. The idea of sharing computational resources between basic tasks and task transformations also connects to the idea of *homoiconicity* from programming language theory—a homoiconic programming language is one in which programs can be manipulated within the language just as data can. The task representations I propose are like programs for performing tasks, and thus using the same mechanism for transforming task representations that is used for performing them therefore makes the system homoiconic. These connections may provide interesting directions for future research.

On language and reasoning: Throughout the dissertation, I show that task transformation offers a better inductive bias for learning than approaches based on only a language description of a task. In fact, task transformation can be useful for adapting a task representation constructed from language. That is, language alone may not be a sufficient representational system. Instead, it may be only one piece of the puzzle. In future work, architectures like the one I proposed could provide a basis for exploring the interactions between language and reasoning over development, and how the connection between language and thought "originates, changes, and grows" (Vygotsky, 1934).

3 Connections to broader issues in cognitive science

Representational redescription: My work was inspired in part by the work of Karmiloff-Smith (1986); Clark and Karmiloff-Smith (1993) on *representational redescription*. In particular, Clark & Karmiloff-Smith propose that in human cognition, our internal representations become "objects for further manipulation." This is precisely the idea underlying my approach to task transformation—the task representations become objects for explicit manipulation. Thus, exploring whether architectures and algorithms like those I proposed could model some of the phenomena they considered would be an exciting direction for future work.

Compositional symbol manipulation: My work also shows that some features are not necessary to achieve some flexible behvaior. In particular, my algorithm does not build in explicitly compositional representations of tasks, as prior work has suggested is necessary for adaptation (Lake and Baroni, 2018, e.g.). Nevertheless, my model can learn to exploit the shared structure in the concept of "losing" across a few card games to achieve 85% performance in losing a game it has never tried to lose before. Similarly, it can achieve perfect adaptation to held-out visual concepts via trained meta-mappings, and near-perfect adaptation from held-out meta-mappings. Hard-coded compositional structure does not appear necessary to achieve effective adaptation.

Furthermore, there are a number of potential benefits to letting the compositional structure emerge rather than building it in. First, the structure does not need to be hand-engineered specially for each domain. Our system required no special knowledge about the domains beyond the basic tasks and the relationships between them. The fact that some of these relationships corresponded to e.g. permutations of variables in the polynomial domain did not need to be hard-coded, instead the model was able to discover it from the patterns of the mappings (as indexed by its ability to generalize well to held-out permutations).

The second advantage of letting compositionality emerge is that it can potentially allow for novel decompositions at test time. The ability of our model to perform well on held-out meta-mappings supports this hope. Furthermore, the ability of the model to extrapolate a meta-mapping learned on color tasks to shape tasks in the RL domain provides further promising evidence. These results are suggestive of the ability of my approach to extrapolate beyond what it has experienced with flexibility and systematicity closer to that of human cognition.

Model-based methods: Adaptating to task alterations has also been explored in the context of model-based reinforcement learning. However, in general these approaches require a new reward function for the model to plan over. Task transformation could offer a solution to the problem of obtaining an adapted reward estimator when the environment changes, or adapting a learned world-model to a change in dynamics. Thus, algorithms like those I propose could complement model-based methods.

Cognitive control: My research offers potential connections to cognitive control. First, the task network has "default" weights that are modified by task-specific adjustments. This builds on and extends the ideas about cognitive control considered by Cohen et al. (1990), by providing learned task representations that can module the task-network's default behavior.

I show a simple experiment demonstrating bias towards a more frequent task in the default weights when no task representation is provided. Furthermore, the idea of adaptation as task transformation offers other possible research directions. For example, interference on task switching could perhaps be modeled by an imperfect transformation.

4 Conclusions

In this dissertation, I have proposed a computational approach by which a deep learning system can successfully perform a novel task on its first try. The approach is based on learning task representations, and learning to transform those task representations based on relationships between tasks. This approach does not require building in prior domain knowledge, and so is broadly applicable. I demonstrated my approach across a variety of domains, including card games, video games, visual concepts, and a mathematical domain that I did not discuss in this summary. Across all these settings, my approach was able to achieve 80-90% performance on a novel task on its first try. This adaptation also serves as a useful starting point for later learning, allowing the system to master the new tasks much more efficiently.

This work therefore addresses some of the key critiques raised by cognitive scientists and philosophers about using deep learning approaches as models of human intelligence. My work shows that deep learning models can achieve rapid and flexible reuse of knowledge, without relying on compositional representations or explicit symbol manipulation. It thereby offers a way to build cognitive models that can adapt more flexibly to novel tasks.

My work also has a number of broader implications, which I discuss in the concluding chapter. In brief, my work provides some new perspective on prior ideas about the role of recursion in cognition (Fodor, 2008), the role of language in thought (Vygotsky, 1934), and the rerepresentation of knowledge in cognitive development (Karmiloff-Smith, 1986). It relates to many other areas of cognitive research, such as cognitive control, continual learning and consciousness. In addition, it has potential implications in domains ranging from education (exploring when and how analogies help students learn) to anthropology (exploring how culturally-constructed educational systems shape our cognitive flexibility). Finally, it offers a route toward constructing artificial intelligence systems that have cognitive flexibility a step closer to that exhibited by humans.

References

Baars, B. J. (2005). Global workspace theory of consciousness: Toward a cognitive neuroscience of human experience. *Progress in Brain Research*, 150:45–53.

Clark, A. and Karmiloff-Smith, A. (1993). The Cognizer's Innards: A Psychological and Philosophical Perspective on the Development of Thought. *Mind & Language*, 8(4):487–519.

Cohen, J. D., Dunbar, K., and McClelland, J. L. (1990). On the control of automatic processes: A parallel distributed processing account of the stroop effect. *Psychological Review*, 97(3):332–361.

Dabney, W., Kurth-nelson, Z., Uchida, N., Starkweather, C. K., Hassabis, D., Munos, R., and Botvinick, M. (2020). A distributional code for value in dopamine-based reinforcement learning. *Nature*, (January):1–32.

Fodor, J. A. (2008). *LOT 2: The language of thought revisited*. Oxford University Press on Demand.

Fodor, J. A. and Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.

Karmiloff-Smith, A. (1986). From meta-processes to conscious access: Evidence from children's metalinguistic and repair data. *Cognition*, 23(2):95–147.

Kriegeskorte, N. (2015). Deep neural networks: a new framework for modelling biological vision and brain information processing. *Annual Review of Vision Science*, pages 417–446.

Lake, B. M. and Baroni, M. (2018). Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, pages 1–12.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2017). Building Machines that learn and think like people. *Behavioral and Brain Sciences*, pages 1–55.

Marcus, G. (2018). Deep Learning: A Critical Appraisal. *arXiv preprint*, pages 1–27.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., Van Den Driessche, G., Graepel, T., and Hassabis, D. (2017). Mastering the game of go without human knowledge. *Nature*, 550.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

Turchetta, M., Berkenkamp, F., and Krause, A. (2016). Safe exploration in finite Markov decision processes with Gaussian processes. *Advances in Neural Information Processing Systems*, pages 4312–4320.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., Vezhnevets, A. S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T. L., Gulcehre, C., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Kavukcuoglu, K., Hassabis, D., Apps, C., and Silver, D. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354.

Vygotsky, L. (1934). *Thought and Language*. MIT Press, 1986 edition.

Yamins, D. L. K., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., and DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences of the United States of America*, 111(23):8619–8624.