

Zero-Episode Few-Shot Contrastive Predictive Coding: Solving intelligence tests without prior training

Tomer Barak¹ Yonatan Loewenstein^{1,2}

Abstract

Video prediction models often combine three components: an encoder from pixel space to a small latent space, a latent space prediction model, and a generative model back to pixel space. However, the large and unpredictable pixel space makes training such models difficult, requiring many training examples. We argue that finding a predictive latent variable and using it to evaluate the consistency of a future image enables data-efficient predictions because it precludes the necessity of a generative model training. To demonstrate it, we created sequence completion intelligence tests in which the task is to identify a predictably-changing feature in a sequence of images and use this prediction to select the subsequent image. We show that a one-dimensional Markov Contrastive Predictive Coding (M-CPC_{1D}) model solves these tests efficiently, with only five examples. Finally, we demonstrate the usefulness of M-CPC_{1D} in solving two tasks without prior training: anomaly detection and stochastic movement video prediction.

1. Introduction

Changes in the world are often dominated by the dynamics of a relatively small number of latent variables. Identifying these variables is useful for making predictions. For example, in video prediction, the task is to predict an image from a sequence of its preceding images. To that goal, video prediction models often assume a small number of latent variables and learn to predict them (Liu et al., 2021). However, the learning of the mapping of these latent variables to the pixel space requires the training of a generative model, which requires a large number of examples. The number of

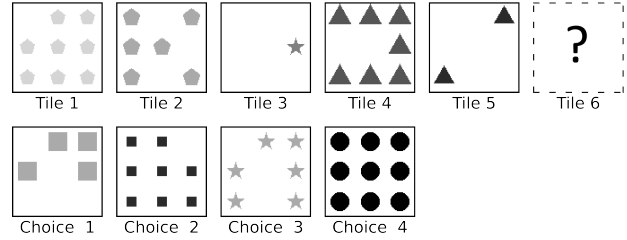


Figure 1. A sequential intelligence test. The images along the sequence become darker. The rest of their characterizing features are random. Only the 4th choice adheres to the shading rule.

examples can be somewhat reduced if the expressivity of the latent encoder is limited (Kumar et al., 2019), or if a simple structure is imposed on the latent variables (Minderer et al., 2019; Kim et al., 2019; Yang et al., 2018).

Our focus here is on a different class of problems, in which the task is to learn a predictive latent model only and use it to evaluate the *consistency* of a given image with its preceding images. Solving this problem enables the identification of incongruent images, or as we focus in this paper, the *selection* of the predictable or congruent image from a set of alternative choices, rather than creating it. We argue that because no generative model training is needed, this problem can be solved using only a small number of examples. In humans, the ability to learn predictive latent variables and use them to select congruent predictions is quantified with intelligence tests, whose score is highly correlated with success in the job market and the academia (Sternberg, 1977; Raven et al., 1998; Lohman, 2000; Kaplan & Saccuzzo, 2009; Siebers et al., 2015). Therefore, we use the framework of intelligence tests to demonstrate the data-efficiency of a latent-guided prediction by selection.

Consider the intelligence test depicted in Fig. 1: five ordered images are presented. The next image, the sixth, is missing. Each image is characterized by features: the number of objects, their color, shape, size, and positions. The images were constructed such that one of the features predictably changes along the sequence according to a simple deterministic rule, while the rest of the features are either constant or randomly changing. Because of the random nature of some of the features, predicting the sixth image exactly is impossible. However remarkably, humans are able to iden-

¹The Edmond and Lily Safra Center for Brain Sciences

²Department of Cognitive Sciences, The Federmann Center for the Study of Rationality, The Alexander Silberman Institute of Life Sciences, The Hebrew University, Jerusalem. Correspondence to: Tomer Barak <tombarak@mail.huji.ac.il>.

tify the image which is congruent with the sequence, out of the four alternative choices depicted in Fig. 1, by finding the predictably changing feature without prior training.

In this paper, we use a Contrastive Predictive Coding (CPC) algorithm (Oord et al., 2018) which was shown useful for finding predictive latent variables (Anand et al., 2019; Henaff, 2020; Yan et al., 2020). This self-supervised algorithm optimizes an infoNCE loss in which consecutive inputs in a sequence are regarded as positive examples and non-consecutive inputs as negative examples. Usually, to get accurate data representations, CPC models are trained over large datasets. However, we hypothesized that the selection task required for solving intelligence tests such as the one depicted in Fig. 1 are solvable by inaccurate data representations and therefore that a CPC algorithm can solve them without *any* prior training. This is an extreme example of a few-shot learning in which the training is done using only the five images of the test, with zero training episodes.

2. Intelligence Tests

Inspired by previous intelligence tests generating algorithms (Wang & Su, 2015; Barrett et al., 2018), sequences of K gray-scale images \mathbf{x}_j (e.g in Fig. 2) were generated in the following way: each image included 1-9 identical objects arranged on a 3×3 grid. An image was characterized by a low dimensional vector of features, \mathbf{f}_j where f_j^i denotes the value of feature i in image j . We used the following five features: the number of objects in an image (possible values: 1 to 9), their shade (6 linearly distributed gray scale values), the shapes (circle, triangle, square, star, hexagon), their size (6 linearly distributed values for the shapes' enclosing circle circumference), and positions (a vector of grid positions that was used to place the shapes in order). The image \mathbf{x}_j was constructed according to its characterizing features by a non-linear and complex generative function $\mathbf{x}_j = g(\mathbf{f}_j)^1$.

One of the features f^p predictably changed along the sequence according to a simple deterministic rule $f_{j+1}^p = u(f_j^p)$ while the other features were either constant over the images or changed randomly (values were i.i.d). We refer to the randomly-changing features as *distractors* and their number is considered a measure of the difficulty of the test. After observing a sequence of K images, the agent's task was to select the correct $K + 1^{\text{th}}$ image from a set of n optional choice images that were generated using the same generative function g from the feature space. In the correct choice, f^p followed the deterministic rule $f_{K+1}^p = u(f_K^p)$, whereas in the incorrect choices it did not follow that rule and was instead randomly chosen from the remaining possible values. The features that were constant or randomly changing in the sequence were also constant or changed

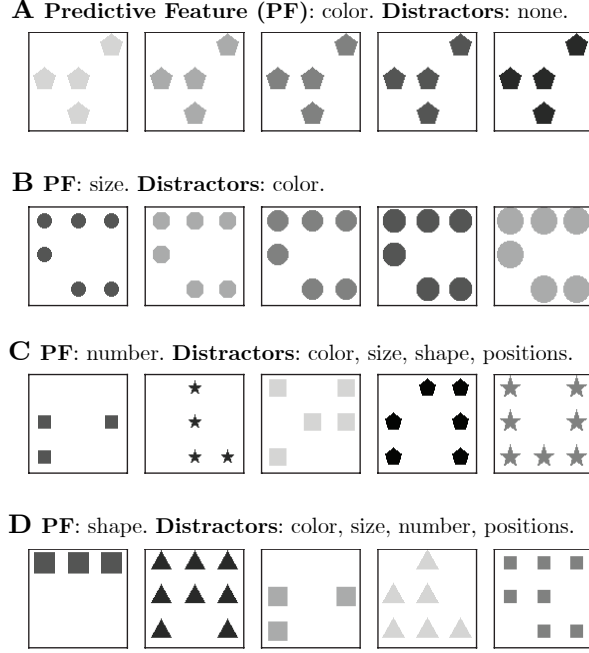


Figure 2. Sequences with various predictive, constant and random features. We call the random features distractors, and their number determines an intelligence test's difficulty.

randomly in all four choices.

3. Markov 1D Contrastive Predictive Coding

Our main challenge in solving the intelligence tests is to find a latent variable that changes in a simple deterministic way along the test's five images sequence. Consider an encoder function Z , a predictor function T , two images \mathbf{x}_a and \mathbf{x}_b and the prediction error

$$\epsilon_{a,b}(Z, T) = \left(T(Z(\mathbf{x}_a)) - Z(\mathbf{x}_b) \right)^2 \quad (1)$$

By construction, for the true encoder and predictor functions $Z^* = g^{-1}$ and $T^* = u$, $\epsilon_{a,b}(Z^*, T^*) = 0$ if a and b are two consecutive images ($b = a + 1$), and $\epsilon_{a,b}(Z^*, T^*) \neq 0$ otherwise.

The challenge is that Z^* and T^* are unknown. However, given a sequence of K ordered images, we can approximate Z^* and T^* by finding Z and T that minimize the prediction error for consecutive images and maximize it for the non-consecutive ones. Formally, we define a contrastive infoNCE loss based on those prediction errors

$$\mathcal{L} = -\frac{1}{K-1} \sum_{j=1}^{K-1} \log \frac{e^{-\epsilon_{j,j+1}}}{\sum_{j'} e^{-\epsilon_{j,j'}}} \quad (2)$$

¹Code for generating intelligence tests available on request.

and find Z and T that minimize it.

The model’s encoder Z and predictor T are implemented by deep neural networks. Because the true predictor is one dimensional, we used a convolutional network for the encoder $Z(\mathbf{x})$ from the 100×100 pixel space to a single neuron; For the predictor, $T(Z(\mathbf{x}))$, we used a residual network $T(Z(\mathbf{x})) = Z(\mathbf{x}) + \Delta T(Z(\mathbf{x}))$ where ΔT is a fully-connected network. This variant of the CPC algorithm predicts a 1D latent variable based only on its most recent value. Thus, we marked it as M-CPC_{1D}.

4. Results

4.1. Solving Tests Without Prior Training

To quantify the performance of M-CPC_{1D} in solving intelligence tests without prior training, we applied it on a multitude of intelligence tests. For brevity, we focused in the main text on a specific predictive feature, the size of the objects, which increased monotonically throughout. Tests in which other features changed predictably are shown in the supplementary material. The 4 remaining features were either constant or randomly changing, resulting in a total of $2^4 = 16$ test conditions (Fig. 3).

Each intelligence test was solved in the following way: First, we randomly initialized the networks corresponding to Z and T . We then updated these networks’ weights with a single optimization step in the direction of minimizing the loss function (Eq. 2)². After the optimization step, we selected the choice image that had the lowest prediction error out of the four choices as the answer of this intelligence test.

Remarkably, we found that training with only the $K = 5$ images that are given within the tests is sufficient for solving easy tests, as well as for achieving a substantially higher than chance performance in the more difficult tests (Fig. 3).

4.2. Meta-Learning Sample Complexity

To evaluate the potential benefit of prior training and obtain the sample complexity of M-CPC_{1D}, we tested the performance of models whose parameters are learned, rather than chosen randomly. Specifically, we performed an episode-based meta-learning, in which prior to performing a certain few-shot learning task, a model is trained on episodes (one optimization step per episode) that are random realizations of the same few-shot learning task (Thrun & Pratt, 1998; Sung et al., 2018). This was done for the 16 test conditions of Fig. 3.

²We found that a single gradient step achieved comparable results to a full minimization of the loss with more steps. We used the RMSprop optimizer with learning rate $\eta = 4 \cdot 10^{-4}$.

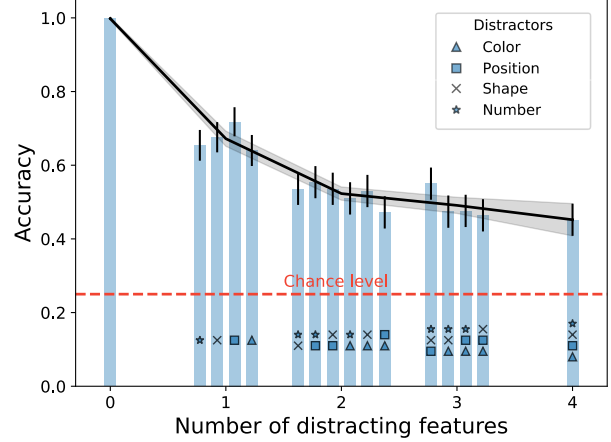


Figure 3. Performance Without Prior Training. Accuracy in 16 test conditions in which the predictive feature was the objects size that increased along the sequences. The remaining 4 features were either distractors (marked according to the legend) or constant (not marked). Performance was evaluated using 500 randomly-generated intelligence tests (see section 2 for details). Error bars correspond to 95% confidence intervals. The black line and its shade are the average accuracy per difficulty and the corresponding standard deviation.

As depicted in Fig. 4, meta-learning improved the performance of M-CPC_{1D} to above 85% accuracy in all test conditions within several hundreds of training episodes.

4.3. Cross-Domain Generalization

Meta-learning algorithms often overfit to the tasks they were trained on, impairing their cross-domain generalization (Li et al., 2017; Yin et al., 2020; Rajendran et al., 2020). To evaluate the cross-domain generalization properties of M-CPC_{1D} we extensively trained (1000 episodes) networks on intelligence tests with certain feature rules, and then tested them with intelligence tests that were characterized by other feature rules. Specifically, we trained and tested the networks using intelligence tests in which the predictive feature was either the size or the color of the objects (increasing monotonically), and the rest of the features could either be all constant (easy condition), or all distracting (hard condition). In total, we considered 4 types of intelligence tests: size-easy, size-hard, color-easy, color-hard; and we crossed between them in the training and final-evaluation stages of the model.

The emerging picture is interesting (Fig. 5). Within the same predictive feature, we find that training using easy episodes is more effective than training with hard ones. Interestingly, after training with the easy episodes, networks’ performance in the hard tests was comparable to their performance in the easy ones. These results suggest that the difference in the asymptote performances in Fig. 4 in the different conditions

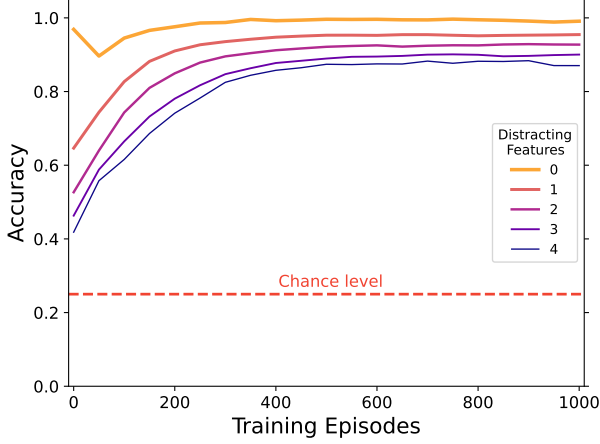


Figure 4. Meta-Learning Sample Complexity. For each of the 16 test conditions in Fig. 3, we trained the model using 1000 training episodes. Performance was evaluated every 50 training episodes. Accuracy measure corresponds to the average performance of 50 randomly-initialized models, each evaluated using 500 intelligence tests that were randomly generated from the same test condition. For clarity, performance is also averaged over test conditions that share the same number of distractors (thus, for example, 0 distractors correspond to a single test condition whereas 2 distractors depict the average over 6 test conditions).

may reflects differences in the training episodes rather than differences in the difficulty of the tests. In other words, after training, the networks can solve even the most difficult tests if these tests are preceded by training using easy episodes.

When considering training and testing on different predictive features, the picture is more complex. Training with easy episodes of one feature rule *improved* performance in hard tests of the other feature rule, but was *detrimental* to performance when the tests of the other rule were easy. Training with hard episodes, on the other hand, was catastrophic, bringing performance in both easy and hard tests of the other predictive feature to chance levels.

4.4. Conclusion

Our main result is that M-CPC_{ID} can successfully solve intelligence tests without any prior training. Moreover, training the model by meta-learning can either improve or impair the performance of the model, depending on the feature alignment between the training and testing domains. These results indicate that zero-episode few-shot training can outperform trained models in environments in which domain shifts are expected, demonstrating the potential benefit of few-shot learning without any prior training.

	Training sequences					Accuracy
	Size (easy)	Size (hard)	Color (easy)	Color (hard)	None	
Size (easy)	98.5% ±0.2%	89.5% ±0.4%	78.8% ±0.5%	25.2% ±0.5%	96.8% ±0.2%	100% 75% 50% 25% 0%
Size (hard)	98.1% ±0.2%	88.3% ±0.4%	77.0% ±0.5%	25.6% ±0.5%	41.9% ±0.6%	
Color (easy)	61.5% ±0.6%	24.7% ±0.5%	100.0% ±0.0%	93.0% ±0.3%	96.6% ±0.2%	
Color (hard)	65.2% ±0.6%	25.2% ±0.5%	100.0% ±0.0%	93.7% ±0.3%	55.1% ±0.6%	

Figure 5. Cross-Domain Generalization. Randomly initiated networks Z and T were trained for 1000 episodes on a certain intelligence test type, and then evaluated over the 4 intelligence test types with 500 tests each. The right column presents the performance without prior training for comparison. Accuracies presented are averaged over 50 experiments and shown with 95% confidence intervals.

5. Applications

5.1. Stochastic Movement Prediction

To demonstrate the usefulness of M-CPC_{ID}, we applied it to a video prediction task. The video we chose to predict is similar to those of the Stochastic Movement (SM) dataset (Babaeizadeh et al., 2017), in which a random shape moves in a random direction from the middle of the frame to one of its sides. It is difficult to train effective deterministic video prediction models in stochastic environments such as the SM. Therefore, stochastic video prediction models have been used to make predictions in such settings (Babaeizadeh et al., 2017; Kumar et al., 2019). Our approach, by contrast, was to use the data-efficiency of M-CPC_{ID} to predict a video with only the first five frames of that video, without training on other stochastic videos.

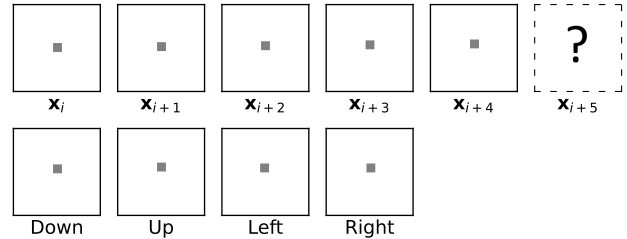


Figure 6. Stochastic Movement as Intelligence Tests. A video of a 10x10 pixel square moving upwards is predicted by treating it as an “intelligence test”. Five consecutive frames of the video are used to predict the sixth frame, whose identity is chosen out of 4 images that are shifted by one pixel in one of the four directions relative to the last frame.

Without loss of generality, we evaluated the video prediction ability of M-CPC_{1D} with one video in which a square moved upwards (Fig. 6). We optimized M-CPC_{1D} on the first $K = 5$ frames $\{\mathbf{x}_i\}_{i=1}^K$ of the video and used the latent variable to select the sixth frame $\tilde{\mathbf{x}}_{K+1}$ out of four possible motion directions, similar to the way we solved the intelligence tests. We found that based on five frames, the model can correctly predict the $K + 1$ frame with a probability of $97\% \pm 1\%$.

To predict the next frame ($K + 2$), we used the last $K - 1$ frames of the original video and the predicted frame $\tilde{\mathbf{x}}_{K+1}$ to create a new sequence of length K ($\{\mathbf{x}_i\}_{i=2}^K \cup \{\tilde{\mathbf{x}}_{K+1}\}$). We then trained new, randomly-initialized networks Z and T , on the new sequence and used it to predict frame $K + 2$, and so on. We iterated this process for 45 frames and the results are depicted in Fig. 7.

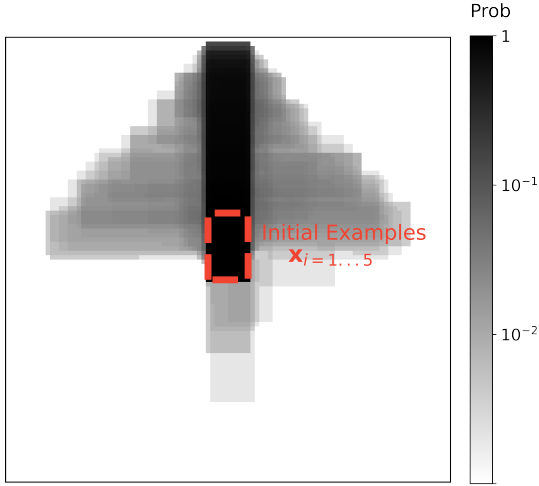


Figure 7. **Predicted Videos.** 500 videos obtained by video prediction with M-CPC_{1D}, conditioned on the 5 initial example frames. A pixel’s shade corresponds to the ratio of predicted videos, in logarithmic scale, that visited that pixel out of the 500 videos.

As demonstrated in Fig. 7, the first $K = 5$ frames and the iterative process are enough for M-CPC_{1D} to predict the video correctly with a high probability.

5.2. Anomaly Detection

To solve intelligence tests, we used a predictive latent variable to *select* a congruent image from a set of alternative choice images. Predictive latent variables can also be used for anomaly detection in tasks that do not entail a selection between alternatives. Specifically, given a sequence of images, the task is to determine whether the last image is congruent or incongruent with its preceding images.

Consider the two sequences depicted in Fig. 8. The task is to determine, without prior training, that there is no anomaly

in the top sequence of images (tile 6 is congruent with its preceding tiles) while there is an anomaly in the bottom sequence (tile 6 is incongruent with its preceding tiles). To classify the congruency of a candidate image \mathbf{x}_c with a given sequence, we performed a single optimization step on the five sequence images with our loss function (Eq. 2). We then compared the prediction error of the candidate image $\epsilon_{K,c}$ to the average of the prediction errors of the sequence’s consecutive images:

$$\epsilon_{th} = \frac{1}{K-1} \sum_{j=1}^{K-1} \epsilon_{j,j+1} \quad (3)$$

We use a threshold parameter α to classify candidate images: when $\epsilon_{K,c} > \alpha \cdot \epsilon_{th}$ we classify the candidate image as anomalous; when $\epsilon_{K,c} < \alpha \cdot \epsilon_{th}$ we classify it as congruent.

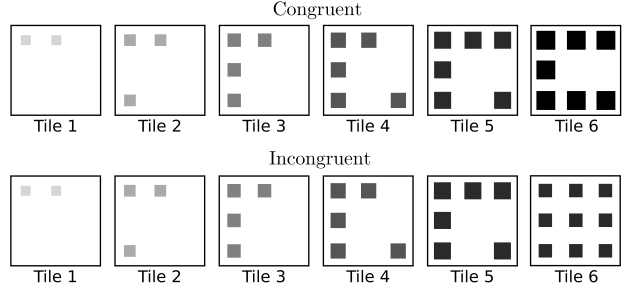


Figure 8. **Anomaly Detection Tests.** We provide our model a sequences of five images. Additionally, we generated two candidate images: one congruent and one incongruent with the sequence. The task of the model is to classify whether a candidate image is congruent or not, based only on the sequence, without comparing the two candidates.

Image sequences of length $K = 5$ were created such that the size, number and shade of the objects in the images increased monotonically along the sequences, while the objects’ shape and the order of grid placement were constants. Two images were candidates for anomaly in each sequence: a congruent image with features that changed according to the sequence rules, and an incongruent image in which the size, number and color of the shapes did not follow the sequence rules. The shape and order of grid placement was also constant in the incongruent image. To evaluate the anomaly detection performance, we generated 500 such tests. With this classification criterion, we achieved a success rate of $85\% \pm 3\%$ (taking into account false positive and misses results; chance accuracy is 50%). Remarkably, this result is achieved without any prior training, using only the five images of each sequence, and without relying on selection from alternatives.

6. Conclusion

We showed that an easy-to-train latent prediction model M-CPC_{1D} can successfully solve prediction tasks. Specifically, we used the predictive latent variable to evaluate the consistency of an image with a sequence of preceding images by training on those preceding sequence images alone. This consistency evaluation allowed us to solve three tasks without prior training: 1) Intelligence tests, in which the task is to select the image that is most consistent with its preceding images. 2) Video prediction in which we showed that a small number of video frames, together with an iterative process, are sufficient to make some predictions in a simple video. 3) And an anomaly detection task. Common to all these examples is that reasonable performance is achieved with an extremely small number of examples. This highlights the data efficiency of latent predictions.

It is generally argued that one main difference between human and machine learning is the amount of data required for the learning. For example, GPT-3 has been trained on hundreds of billions of words, more than three orders of magnitude more words than humans use when learning to speak (Hart & Risley, 1995). Our results demonstrate that also in machine learning, data efficiency can go a long way.

7. Relation to Previous Works

Intelligence tests solvers - Human intelligence is often measured using intelligence tests that are similar to ours. Solving intelligence tests relies on finding relevant features in the provided examples and the rules that govern these features (Blum & Blum, 1975; Siebers et al., 2015). Traditional computational models that solved intelligence tests utilized either prior knowledge of the relevant features (Rasmussen & Eliasmith, 2011), knowledge about the rules that govern these features (Sun & Dai, 2018), or both (Carpenter et al., 1990). Today, machine learning models are able to learn the relevant features and rules using deep artificial neural networks (Barrett et al., 2018; Hill et al., 2019). However, they relied on both supervised learning and large datasets, unlike humans that seem to be able to solve such tests without prior training. Other models solved intelligence tests with unsupervised learning (Zhuo & Kankanhalli, 2020) and by meta-learning (Santoro et al., 2017; Kim et al., 2020), but they also relied on extensive prior training before solving the tests. Our work presents an ability to solve intelligence tests without any prior training.

Contrastive learning - Contrastive loss function are widely used for self-supervised learning (Chen et al., 2020; Le-Khac et al., 2020). One contrastive algorithm, the CPC algorithm, (Oord et al., 2018) is used for finding predictive latent representations, which is useful for data-efficient image recognition (Henaff, 2020) and learning world-models

that supports robotic object manipulation (Yan et al., 2020) and playing Atari games (Anand et al., 2019). Compared with these works, which relied on extensive training, we trained M-CPC_{1D} with only five images. This was possible because we used a 1D latent variable feed-forward encoder rather than a higher-dimensional recurrent network encoder.

Video prediction models - The video prediction task is useful for various down-stream applications such as representational learning and model-based reinforcement learning. Therefore, many deep learning models were developed to solve this task (Oprea et al., 2020). State-of-the-art models utilize a latent prediction models to solved the task (Minderer et al., 2019; Kim et al., 2019; Yang et al., 2018; Lee et al., 2021). These models also required generative models, which needed training as well. A stochastic model for the latent variable has been shown useful for video generation (Kumar et al., 2019; Franceschi et al., 2020; Babaeizadeh et al., 2017). However, as we show here, a deterministic latent model is sufficient for the selection task even in stochastic environments, which allows for better data efficiency.

Relation networks and meta-learning - M-CPC_{1D} is similar to Relation Network (RN) (Sung et al., 2018), a model which utilized episode based meta-learning for classifying examples by the abstract relation between them. While RN can learn general abstract relations between inputs, our model is inductively biased for learning a future-directed residual relation between consecutive latent representations. As a result of this and the fact that the task is a selection task, RN models require a large number of episodes for training while the intelligence tests can be solved without any prior training.

References

- Anand, A., Racah, E., Ozair, S., Bengio, Y., Côté, M.-A., and Hjelm, R. D. Unsupervised State Representation Learning in Atari. (NeurIPS), 2019. URL <http://arxiv.org/abs/1906.08226>.
- Babaeizadeh, M., Finn, C., Erhan, D., Campbell, R. H., and Levine, S. Stochastic Variational Video Prediction. October 2017. URL <http://arxiv.org/abs/1710.11252>.
- Barrett, D. G. T., Hill, F., Santoro, A., Morcos, A. S., and Lillicrap, T. Measuring abstract reasoning in neural networks. 2018. ISSN 17740746. doi: 10.1051/agro/2009059. URL <http://arxiv.org/abs/1807.04225>. ISBN: 1807.04225v1.
- Blum, L. and Blum, M. Toward a mathematical theory of inductive inference. *Information and Control*, 28

- (2):125–155, 1975. ISSN 00199958. doi: 10.1016/S0019-9958(75)90261-2.
- Carpenter, P. A., Just, M. A., and Shell, P. What One Intelligence Test Measures: A Theoretical Account of the Processing in the Raven Progressive Matrices Test. (3): 28, 1990.
- Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 1597–1607. PMLR, November 2020. URL <https://proceedings.mlr.press/v119/chen20j.html>. ISSN: 2640-3498.
- Franceschi, J.-Y., Delasalles, E., Chen, M., Lamprier, S., and Gallinari, P. Stochastic Latent Residual Video Prediction. *arXiv:2002.09219 [cs, stat]*, August 2020. URL <http://arxiv.org/abs/2002.09219>. arXiv: 2002.09219 version: 4.
- Hart, B. and Risley, T. R. The early catastrophe: The 30 million word gap by age 3. *American Educator*, 27(1): 4–9, 1995.
- Henaff, O. Data-Efficient Image Recognition with Contrastive Predictive Coding. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 4182–4192. PMLR, November 2020. URL <https://proceedings.mlr.press/v119/henaff20a.html>. ISSN: 2640-3498.
- Hill, F., Santoro, A., Barrett, D. G. T., Morcos, A. S., and Lillicrap, T. Learning to Make Analogies by Contrasting Abstract Relational Structure. 2019. URL <http://arxiv.org/abs/1902.00120>.
- Kaplan, R. M. and Saccuzzo, D. P. *Psychological Testing: Principles, Applications, and Issues*. 2009. ISBN 0-495-09555-9.
- Kim, Y., Nam, S., Cho, I., and Kim, S. J. Unsupervised Keypoint Learning for Guiding Class-Conditional Video Prediction. October 2019. URL <https://arxiv.org/abs/1910.02027v1>.
- Kim, Y., Shin, J., Yang, E., and Hwang, S. J. Few-shot Visual Reasoning with Meta-analogical Contrastive Learning. *arXiv:2007.12020 [cs, stat]*, July 2020. URL <http://arxiv.org/abs/2007.12020>. arXiv: 2007.12020.
- Kumar, M., Babaeizadeh, M., Erhan, D., Finn, C., Levine, S., Dinh, L., and Kingma, D. VideoFlow: A Conditional Flow-Based Model for Stochastic Video Generation. March 2019. URL <http://arxiv.org/abs/1903.01434>.
- Le-Khac, P. H., Healy, G., and Smeaton, A. F. Contrastive Representation Learning: A Framework and Review. *IEEE Access*, 8:193907–193934, 2020. ISSN 2169-3536. doi: 10.1109/ACCESS.2020.3031549. Conference Name: IEEE Access.
- Lee, W., Jung, W., Zhang, H., Chen, T., Koh, J. Y., Huang, T., Yoon, H., Lee, H., and Hong, S. Revisiting Hierarchical Approach for Persistent Long-Term Video Prediction. April 2021. URL <http://arxiv.org/abs/2104.06697>.
- Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. Learning to Generalize: Meta-Learning for Domain Generalization. *arXiv:1710.03463 [cs]*, October 2017. URL <http://arxiv.org/abs/1710.03463>. arXiv: 1710.03463.
- Liu, B., Chen, Y., Liu, S., and Kim, H.-S. Deep Learning in Latent Space for Video Prediction and Compression. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 701–710, June 2021. doi: 10.1109/CVPR46437.2021.00076. ISSN: 2575-7075.
- Lohman, D. F. Complex Information Processing and Intelligence. In Sternberg, R. J. (ed.), *Handbook of Intelligence*, pp. 285–340. Cambridge University Press, Cambridge, 2000. ISBN 978-0-521-59648-0. doi: 10.1017/CBO9780511807947.015.
- Minderer, M., Sun, C., Villegas, R., Cole, F., Murphy, K., and Lee, H. Unsupervised Learning of Object Structure and Dynamics from Videos. June 2019. URL <http://arxiv.org/abs/1906.07889>.
- Oord, A. v. d., Li, Y., and Vinyals, O. Representation Learning with Contrastive Predictive Coding. 2018. URL <http://arxiv.org/abs/1807.03748>.
- Oprea, S., Martinez-Gonzalez, P., Garcia-Garcia, A., Castro-Vargas, J. A., Orts-Escolano, S., Garcia-Rodriguez, J., and Argyros, A. A Review on Deep Learning Techniques for Video Prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020. ISSN 1939-3539. doi: 10.1109/TPAMI.2020.3045007. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.
- Rajendran, J., Irpan, A., and Jang, E. Meta-Learning Requires Meta-Augmentation. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 5705–5715. Curran Associates, Inc., 2020. URL <https://proceedings.neurips.cc/paper/2020/file/3e5190eeb51ebe6c5bbc54ee8950c548-Paper.pdf>.

- Rasmussen, D. and Eliasmith, C. A neural model of rule generation in inductive reasoning. *Topics in Cognitive Science*, 3(1):140–153, 2011. ISSN 17568757. doi: 10.1111/j.1756-8765.2010.01127.x.
- Raven, J., Raven, J. C., and Court, J. H. *Manual for Raven’s progressive matrices and vocabulary scales*. Pearson, San Antonio, TX, 1998. ISBN 978-0-15-868643-1 978-0-15-468622-0 978-0-15-468623-7 978-1-85639-022-4. OCLC: 697438611.
- Santoro, A., Raposo, D., Barrett, D. G. T., Malinowski, M., Pascanu, R., Battaglia, P., and Lillicrap, T. A simple neural network module for relational reasoning. *Advances in Neural Information Processing Systems*, 2017-Decem: 4968–4977, 2017. ISSN 10495258.
- Siebers, M., Dowe, D. L., Schmid, U., Hernández-Orallo, J., and Martínez-Plumed, F. Computer models solving intelligence test problems: Progress and implications. *Artificial Intelligence*, 230:74–107, 2015. ISSN 00043702. doi: 10.1016/j.artint.2015.09.011. URL <http://dx.doi.org/10.1016/j.artint.2015.09.011>. ISBN: 9780999241103 Publisher: Elsevier B.V.
- Sternberg, R. J. Component processes in analogical reasoning. *Psychological Review*, 84(4):353–378, 1977. ISSN 0033295X. doi: 10.1037/0033-295X.84.4.353.
- Sun, R. and Dai, D. Y. Deep Learning of Raven’s Matrices. *Advances in Cognitive Systems*, pp. 1–6, 2018. URL http://www.cogsys.org/papers/ACS2017/ACS_2017_paper_23_Mekik.pdf.
- Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H. S., and Hospedales, T. M. Learning to Compare: Relation Network for Few-Shot Learning. pp. 1199–1208, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Sung_Learning_to_Compare_CVPR_2018_paper.html.
- Thrun, S. and Pratt, L. Learning to Learn: Introduction and Overview. In Thrun, S. and Pratt, L. (eds.), *Learning to Learn*, pp. 3–17. Springer US, Boston, MA, 1998. ISBN 978-1-4615-5529-2. doi: 10.1007/978-1-4615-5529-2_1. URL https://doi.org/10.1007/978-1-4615-5529-2_1.
- Wang, K. and Su, Z. Automatic generation of Raven’s progressive Matrices. *IJCAI International Joint Conference on Artificial Intelligence*, 2015-Janua(Ijcai):903–909, 2015. ISSN 10450823. ISBN: 9781577357384.
- Yan, W., Vangipuram, A., Abbeel, P., and Pinto, L. Learning Predictive Representations for Deformable Objects Using Contrastive Estimation. 2020. URL <http://arxiv.org/abs/2003.05436>.
- Yang, C., Wang, Z., Zhu, X., Huang, C., Shi, J., and Lin, D. Pose Guided Human Video Generation. July 2018. URL <https://arxiv.org/abs/1807.11152v1>.
- Yin, M., Tucker, G., Zhou, M., Levine, S., and Finn, C. Meta-Learning without Memorization. *arXiv:1912.03820 [cs, stat]*, April 2020. URL <http://arxiv.org/abs/1912.03820>. arXiv: 1912.03820.
- Zhuo, T. and Kankanhalli, M. Solving Raven’s Progressive Matrices with Neural Networks. *arXiv:2002.01646 [cs]*, February 2020. URL <http://arxiv.org/abs/2002.01646>. arXiv: 2002.01646.

Supplementary Materials

S8. Results of other predictive features

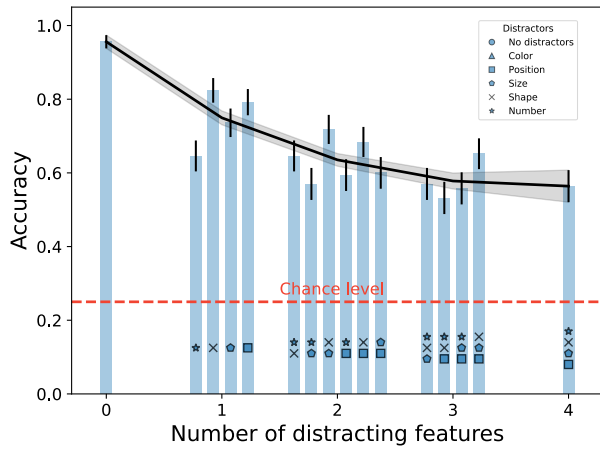


Figure S9. **Predictive Feature: color.** Zero-episodes performance when the color of the shapes became darker along the sequences.

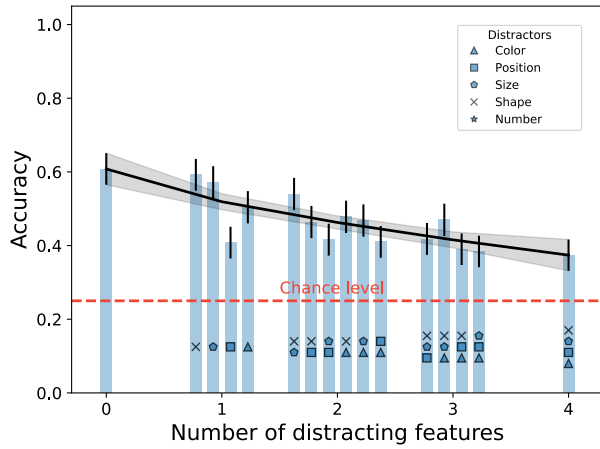


Figure S10. **Predictive Feature: number.** Zero-episodes performance when the number of shapes increased monotonically along the sequences.

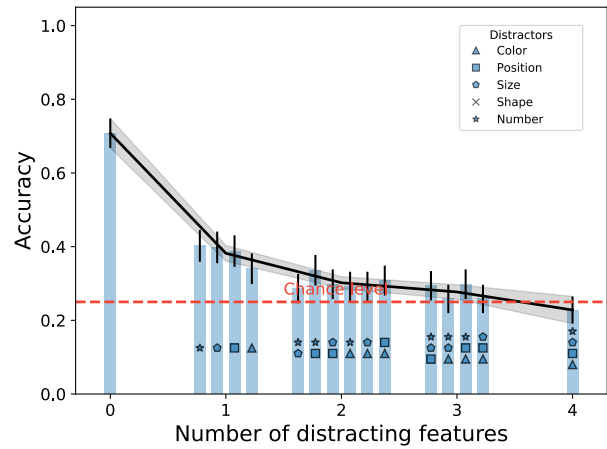


Figure S11. **Predictive Feature: shape.** Zero-episodes performance when the shapes alternated between a triangle and a square along the sequences.