Proceedings of the Twelfth Annual Conference on Advances in Cognitive Systems ACS-2025 (105-124)

Flexible Segmentation for Rule Learning over Object Representations in Abstract Visual Reasoning

Johannes Langer JOHANNES.LANGER@UNI-BAMBERG.DE
Ute Schmid UTE.SCHMID@UNI-BAMBERG.DE
Kognitive Systeme, Otto-Friedrich-Universität Bamberg. An der Weberei 5, 96047 Bamberg, Germany

Abstract

The ability to reason in abstract domains is a key component of human intelligence, and is tested in many intelligence tests such as Raven's Progressive Matrices (RPM). We present ECo-NSR, a novel neuro-symbolic system which aims to explicitly separate perception and problem solving by sequentially layering instance segmentation, feature extraction and analogical reasoning. By separating these three components, the model additionally requires only segmentation masks as labels for training and can continuously improve by exchanging components for new developments in the fields of image segmentation, representation learning and symbolic reasoning. While end-to-end inference with the proposed system is currently not yet possible, our intermediary results indicate the viability of the proposed system and its transferability to other abstract visual reasoning domains.

1. Introduction

Abstract visual reasoning (AVR) is a core ability of human problem solving. It is assessed in many intelligence tests, such as Raven's Progressive Matrices (RPM) (Raven & Court, 1938), which have been used as a measure for human cognitive ability for over 80 years (Raven, 2000; Carpenter et al., 1990). RPM problems require of the solver to identify the missing ninth panel in a 3x3 grid from a set of options based on a rule which must be inferred from the eight existing panels in the grid (see Figure 1).

Visual analogy problems also have been addressed with different approaches in AI research (Hernández-Orallo et al., 2016), starting with Evan's analogy system (Evans, 1964) and leading to current work on the abstract reasoning challenge (Chollet, 2019; Chollet et al., 2024). On RPM benchmarks, current AI systems often achieve super-human performance in terms of accuracy (Wu et al., 2020; Shah et al., 2022; Zhao et al., 2023; Malkinski & Mandziuk, 2022), but they achieve their solutions very differently than humans would. These systems rely on a large dataset of labeled RPM problems from which they learn *relational classifiers*, which can then be used to identify the presence of a previously learned relation between known attributes in an unseen sample. Conversely, humans are able to identify relevant attributes and invent new rules on the spot by discovering patterns in the premises and performing analogical reasoning to choose the correct option out of a given set. This highlights an important limitation, which every system above shares: They are

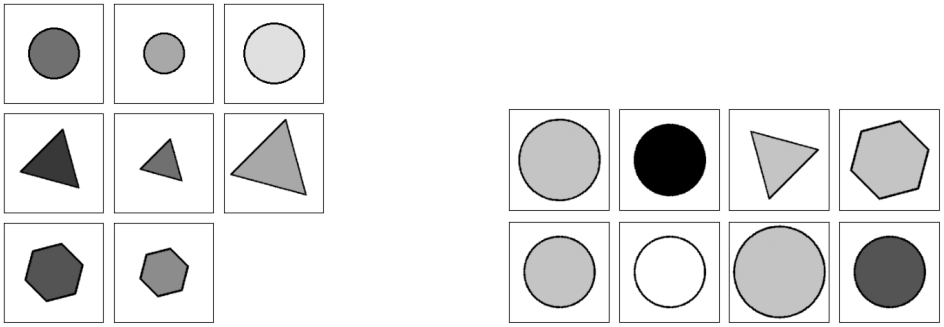


Figure 1: An exemplary progressive matrix from the RAVEN-FAIR (Benny et al., 2021) dataset.

highly dataset dependent, and do not generalize well to problems which contain unseen relations, attributes, and even attribute values.

Designing a solver for RPMs which is not held back by these limitations requires the relations to be learned at inference time rather than at training time, drastically reducing the amount of data available to learn the relation. Inductive Logic Programming (ILP) (Muggleton, 1991) is a symbolic machine learning approach which excels at learning relations from very limited data, but cannot directly be applied to the high-dimensional input which are images of matrices. We propose a framework which allows to use such methods for inference relying only on image data, the **E**xplicit **Co**gnition **N**euro-**S**ymbolic **R**easoner (ECo-NSR). It makes use of object detection, allowing us to conceptually ignore different matrix configurations (see Figure 2). The framework then constructs a symbolic representation from the shapes' coordinate locations in the panels and clustering over per-shape unsupervised feature extraction using the representations for all shapes in the matrix.

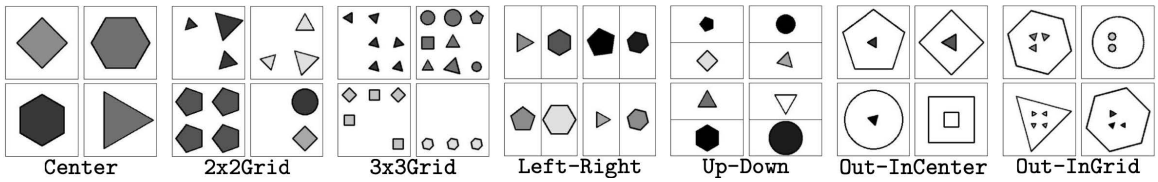


Figure 2: Examples of the seven configurations contained in datasets of the RAVEN family (Zhang et al., 2019) as presented by the authors.

The contributions of this paper can be summarized as follows:

1. We propose a modular framework and design principles for solving AVR problems such as RPMs which addresses important limitations of existing systems.
2. We enable the use of symbolic ILP on image data by building a structural model from clustered continuous representations.
3. We communicate diagnostics of our first implementation of the framework to help future research improve on our design.

2. Related Work

Computational approaches for solving AVR problems address the impressive human ability of analogical reasoning in visual domains. To realize this human ability, computational models need to be able of flexible segmentation of visual scenes together with inference of abstract rules which generalize over the structural commonalities of the given information to generate a solution. In the following, we introduce core aspects of human analogical reasoning, image segmentation, representation learning and inductive logic programming as foundation for our approach. Furthermore, we summarize previous work on solving RPM problems.

2.1 AVR

AVR is a domain of problems which require formulating analogies based on some visual premise (Malkinski & Mandziuk, 2023). For machines, these problems generally require reasoning over a limited set of relevant features extracted from high-dimensional input (i.e. images). This work focuses on a subset of AVR problems in which the analogies are formulated over the attributes of individual shapes, such as RPMs (Raven & Court, 1938; Zhang et al., 2019; Santoro et al., 2018; Benny et al., 2021) or Bongard problems (Bongard, 1970; Nie et al., 2020). These are distinct from the Abstraction and Reasoning Corpus (ARC) (Chollet, 2019), where the reasoning premises are low-resolution pixel grids. As ARC does not involve reasoning over shape attributes, ECo-NSR is not suitable for these problems particularly as it cannot rely on its segmentation approach.

2.2 Human Analogical Reasoning

One of the most influential cognitive theories of analogy is Gentner's structure mapping theory. In this theory, domains and situations are psychologically viewed as systems of objects, object-attributes and relations between objects (Gentner, 1983). An analogy describes a mapping from a base B to a target T. More specifically, objects from B are mapped to objects from T in a way which *maintains the relationships* between these objects (structure preserving mapping). For this mapping to be possible, the reasoner needs to be familiar with the relationships in the base to be able to map them on to the target. In an analogy, the base is used to generate inferences about the target, that is, formulating an analogy is an inductive reasoning process (Holyoak, 2012).

In the context of structure mapping, analogical reasoning has mostly been researched for symbolic problem representations, often for structural descriptions of physics problems such as the Rutherford analogy (Gentner, 1983). The Structure Mapping Enginge (SME) (Falkenhainer et al., 1989) restricts mapping to identical relations which are explicitly represented for base and target. That is, it is not possible to map a relation such as 'greater(a,b)' in the base to a relation 'larger(c,d)' in the target. The LISA model (Hummel & Holyoak, 2019) allows mapping of different relations based on neuro-symbolic approach where predicates and their arguments are mapped based on patterns of activation distributed over semantic primitives. In an experiment designed for comparing different mapping strategies, Wiese et al. (2008) could show that humans map as much structure as possible between base and target and not only that part of information necessary to solve a specific task. Such a strategy might involve mapping of different relations as well as re-representation of the base problem as proposed in the Analogy via Abstraction (AvA) model (Weller & Schmid, 2006).

The human ability to flexibly re-represent problems to construct complex analogical relations has been addressed early on by (Mitchell, 1993) and (Hofstadter & Group, 1995). However, their Copycat model is restricted to letter string analogies. In contrast to visual reasoning problems, for letter strings, the primitives over which re-representation occur are a pre-defined finite set, namely the letters of the alphabet.

2.3 Image Segmentation

Image Segmentation is the umbrella term for combining classification with localization in images (Minaee et al., 2022). Object detection and instance segmentation specifically are two learning tasks which are relevant to this work, as they deal with identifying objects in images at different fidelity. Where object detection is concerned with merely predicting bounding boxes and class membership of objects, instance segmentation is the task of predicting pixel-accurate segmentation masks of these objects.

A more recent development in image segmentation are large, pre-trained, general segmentation models (*foundation models for image segmentation*), most notably Segment Anything (SAM) and its successor (Kirillov et al., 2023; Ravi et al., 2025). The big difference to Mask-RCNN is that SAM models aren't trained on a problem-specific dataset, but pre-trained on a very large and versatile dataset (11 million images with a total of 1 billion masks in the case of SAM). The downside of these large models is of course sustainability concerns due to high computational cost.

2.4 Representation Learning

The presented implementation of ECo-NSR relies on Variational autoencoders (VAEs) (Kingma & Welling, 2014) as its representation model. VAEs as an architecture have been popularized as powerful generative models, but were picked up by the representation learning community due to their ability to learn expressive latent representations in an unsupervised setting. VAEs model the training data $\boldsymbol{X}$ as samples $x^{(i)} \in \boldsymbol{X}$ generated by a true posterior distribution $p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{z})$ based on an underlying random variable $\boldsymbol{z}^{(i)}$, which is essentially the desired representation. The variational posterior $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ approximates the intractable $p_{\boldsymbol{\theta}^*}(\boldsymbol{z}|\boldsymbol{x})$ and is a *recognition model* to infer from a sample $\boldsymbol{x}^{(i)}$ back to the underlying $\boldsymbol{z}^{(i)}$. Note that it doesn't directly produce z but the parameters of a (usually Gaussian) distribution over the possible values of z, from which z can be sampled. There are two directions of extensions to VAEs which are particularly relevant to this work.

First, the Vector-Quantized VAE (van den Oord et al., 2017) and Stochastically-Quantized VAE (Takida et al., 2022) extensions can learn discrete instead of continuous representations by learning a codebook jointly with the model. This is important because the generated latent representations are to be used as input to a symbolic reasoning system, which cannot work with continuous data.

Second, VAEs have been modified to allow the learning of disentangled representations, that is representations where each dimension is associated with exactly one explanatory factor of the underlying data (Bengio et al., 2013; Wang et al., 2022). This is important for at the very least the logical reasoning used in this work, as rules get increasingly large when a visual concept requires many dimensions to be represented as opposed to only one. β-VAE (Higgins et al., 2017) and β-TCVAE (Chen et al., 2018) achieve disentanglement by introducing weighting hyperparameters

to the loss function of VAE. Zhao et al. (2023) propose semantic VAE, which makes use of the detailed labels in the RAVEN dataset to learn a multi-hot representation in a supervised manner. This, however, is very narrow in application due to the high demand on labels.

2.5 ILP

ILP (Muggleton, 1991) belongs to the family of *inductive programming* machine learning approaches, where the goal is to learn computer programs from incomplete specifications such as examples (Flener & Schmid, 2008). In ILP specifically, the learning problem is to learn a hypothesis (program) H which, given background knowledge B, positive examples E^+ and negative examples E^-, is *complete* (entails all positive examples) and *consistent* (entails no negative examples) (Cropper & Dumancic, 2022). One of its great advantages is that it is very data efficient, as modern ILP systems are capable of learning correct (simple) programs from only a single positive example (Hocquette et al., 2024). This data efficiency inspires the use of ILP for finding on-the-fly solutions to RPM problems, as the rules must be inferred from only two example rows.

Recently, the ILP system POPPER (Cropper & Morel, 2021) has become popular in the ILP community. POPPER uses a learning strategy called learning from failures, where a generated hypothesis (using background knowledge and existing constraints) is tested against the training examples. If the hypothesis fails, either by being too general (covering a negative example) or by being too specific (not covering a positive example), generalizations or specifications of the hypothesis may be pruned from the search space, which is considered as new constraints for generating the next hypothesis. The authors prove that if the hypothesis space contains only decidable programs, POPPER is sound, complete and optimal. POPPER has been shown to be able to learn certain programs from only a single positive example, given appropriate bias, background knowledge and a well chosen example (Hocquette et al., 2024). However, an important limitation of POPPER for this work is that the system is unable to learn programs containing constants. This is addressed by the extensions MAGICPOPPER (Hocquette & Cropper, 2023a) and NUMSYNTH (Hocquette & Cropper, 2023b).

2.6 RPM Solvers

Computational Cognitive Models. One of the first computational approaches for RPM problems is a production system proposed by Carpenter et al. (1990). This system relied on hand-coded propositional descriptions of problems and a predefined set of rules over matrix elements. To predict a solution, a rule matching the propositional representation of the input was selected and the predicted answer was compared to the answer choices to choose the best match.

Lovett & Forbus (2017) introduce a cognitive architecture for solving RPM problems which is based on the structure-mapping theory (Gentner, 1983). The proposed model relies on the recognition of qualitative visual concepts, contrasting these concepts within a matrix row (called *pattern of variance*) and finally constructing a structural mapping based on the patterns of variance in the first two matrix rows. This approach stands out from the other discussed solvers as it performs inductive reasoning at inference time. Notably, this is a handcrafted, not a learned model.

Kunda et al. (2013) characterize this type of models as amodal and propositional and argue that humans use imagery-based reasoning strategies to RPM problems. Solving an RPM problem is

rooted in perception and reasoning is based on modal representations of test inputs. They propose an "affine" model, based on iconic visual representations together with affine and set transformations over these representations. However, this model relies on a pre-defined set of pixel-patterns as input. That is, it does not address the ability of humans of flexible segmentation of a given visual pattern based on the given problem context.

Deep Learning Approaches. Wu et al. (2020) introduce the Scattering Compositional Learner (SCL), which approaches solving RPMs by explicitly learning neural networks for representing objects ($\mathcal{N}^o$), attributes ($\mathcal{N}^a$), and relations ($\mathcal{N}^r$). The core idea of this architecture is that – similarly to our work presented in this paper – compositions of these networks are explicitly computed. The composition $\{\mathcal{N}_k^r \circ \mathcal{N}_j^a \circ \mathcal{N}_i^o\}_{ijk}$ indicates whether the k^{th} relationship holds among the j^{th} attribute of the i^{th} object. An MLP calculates a single score from all possible combinations. To solve an RPM problem, the model calculates scores for eight copies of the matrix, each with one of the answer options filled in. The version with the highest score is considered the solution. Note that while SCL does not explicitly perform object detection, the composition of $\mathcal{N}^o$ and $\mathcal{N}^a$ extracts enough features that the symbolic attribute values can be classified using only a single linear layer. Finally, while the model is able to generalize to unseen attribute-relation pairs, a trained model is still constrained to the general relations it was trained on. Malkinski & Mandziuk (2022) report performances of SCL and other deep learning based approaches on RPMs.

Neuro-Symbolic Approaches. Shah et al. (2022) introduce an approach which uses Neural Algorithmic Reasoning (Velickovic & Blundell, 2021) for RPM problems. Their approach involves a two stage process: First, a matrix *row* is embedded using an autoencoder. A high-quality and task-relevant feature extraction is ensured by aligning the representation from an image-based autoencoder with the representation of an autoencoder based on the symbolic labels. Reasoning is performed by a large set of *rule identification networks* for each attribute-relation pair, which is similar to the composition of SCL's $\mathcal{N}^a$ and $\mathcal{N}^r$. However, the number of networks is explicitly based on the number of attribute-relation pairs in the dataset, which also limits the applicability of the system to that exact dataset and only the considered attribute-relation pairs. Furthermore, the set of rule identification networks must be re-trained for each of the seven matrix configurations. Zhao et al. (2023) determine matrix configuration with a learned classifier and extract panel-wise discrete representations using the configuration-appropriate sVAE model. Based on this representation, a matrix-based symbolic reasoning system inspired by cognitive maps (Tolman, 1948) is used. This method also relies on having a learned matrix-representation ready for each relation in the dataset, but is independent from the attribute as it relies on symbolic data.

3. ECo-NSR: Making Cognitive Processes Explicit

ECo-NSR is a framework designed to be able to offer generalized compound models for many different abstract visual reasoning tasks. It achieves this by differentiating itself from the approaches discussed in Section 2.6 in two important ways:

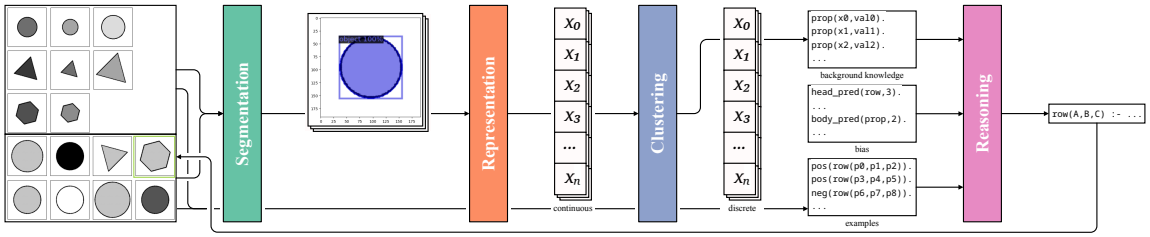


Figure 3: Framework design. During segmentation, shape bounding boxes are produced from panel-wise input. These cropped panels are handed to representation, where they are embedded into a continuous latent space. These embeddings are then clustered per dimension in order to generate the symbolic input to the reasoner. Finally, the reasoner produces a rule from which the correct solution panel follows.

1. Feature extraction operates on shape-level as opposed to panel-level, which alleviates the need for the configuration classifiers required by existing neuro-symbolic approaches (Shah et al., 2022; Zhao et al., 2023).

2. Relation learning happens at inference using ILP as a data-efficient symbolic learning system, which allows the compound model to generalize to completely new relations.

ECo-NSR solves an AVR problem using the following steps: First, all panel images are separately used as inputs to a learned segmentation model. The segmentation model outputs per-shape bounding boxes, which are used to hand versions of the panel images cropped to single shapes to a learned representation model. The representation model outputs a low-dimensional, ideally feature-disentangled continuous representation, which is made discrete by applying a clustering algorithm dimension-wise to the representations from all shapes in the matrix. A detailed symbolic representation is created from the shape coordinates and sizes (as produced by the segmentation model) and per-dimension cluster memberships (as produced by the representation model). From this, the rule governing the particular matrix is learned by the reasoning system. An overview of the framework is shown in Figure 3.

This framework is making cognitive processes explicit in the sense that it uses computer vision, specifically image segmentation, to identify objects in the visual presentation of the problem. According to the structure-mapping theory (Gentner, 1983), knowledge about these objects, their attributes and being able to relate them to each other is a necessary prerequisite for analogical reasoning. By using segmentation before feature extraction and combining this information with the structural information from the problem, we explicitly create these prerequisites and are able to perform inductive, analogical reasoning to solve the problem.

3.1 Segmentation

Using a segmentation model allows ECo-NSR to abstract from the structural composition of images in AVR problems. Not only does this result in the ability to use the same models for each config-

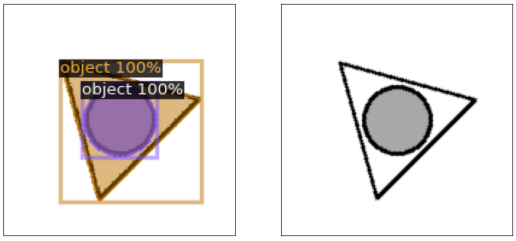


Figure 4: Edge-case example of a RAVEN-F (Benny et al., 2021) configuration `Out-InCenter` panel image. Note that the annotated bounding box of the inner circle overlaps the outer triangle, whereas the annotated mask does not.

uration in the RAVEN-Dataset, but it even lays the groundwork for using the same model *instance* for different types of datasets, as long as the segmentation model can identify the shapes.

Both object detection and instance segmentation can be considered as learning tasks for this model, with considerations to be made regarding the label availability, image features, and performance. Object detection requires only bounding boxes as labels, which involve much less annotation effort. Models also often perform better on object detection than instance segmentation (Minaee et al., 2022). However, depending on the images containing the shapes, there are reasons to prefer instance segmentation over object detection: When shapes aren't presented against a uniform background, crops to bounding boxes contain variation in background, where pixel-exact masked shapes would not. This background may affect the representation model. Additionally, shapes can be overlapping or contained within other shapes. In these cases, it is possible to provide a more uniform shape for the representation model by removing the foreground/inner shapes from the background/outer shapes (shape subtraction). Both bounding boxes and segmentation masks can be used for shape subtraction, but bounding boxes are much more destructive (see Figure 4).

3.2 Representation and Clustering

The goal of the representation step is to produce a low-dimensional disentangled discrete representation. While there may exist data-efficient symbolic reasoners in the future or unknown to us which do not have these requirements, the specific reasoner used in this work requires discrete inputs, and the disentanglement and low dimensionality help reduce the search space of possible rules.

Requiring the representation model itself to directly output discrete representations limits the choice of architectures for this step. Instead, we allow the use of continuous representation models and perform discretization after the fact using clustering over the representations of all shapes in a problem. Not only does this allow the use of e.g. β-TCVAE, but it also results in discrete representations which are based on similarities between the shapes relevant for a specific problem (i.e. local labels), instead of being based on knowledge of possible attribute values in the dataset.

3.3 Reasoning

Importantly, ECo-NSR's reasoner is characterized by **not** being a trained model, but instead being a data-efficient machine learning system which infers the relations, i.e. analogies, directly from the reasoning premise. While this has clear benefits in terms of generalization, it also has the downside that such reasoners must rely on a strong inductive bias to learn the rules governing the analogies. This bias must be designed by hand, and decide which kind of relations theoretically can and cannot be learned.

4. Example Model

4.1 Data

We demonstrate the practical viability of this framework using the RAVEN-F (Benny et al., 2021) dataset. We generate the dataset using the default settings, resulting in a total of 70,000 RPM problems (10,000 in each configuration) with 16 panel images per problem (8 premise panels and 8 answer options). We use this dataset to train both the segmentation and representation models. While fine-grained labels are available due to the dataset being generated, we only use the annotations for bounding boxes and shape masks.

4.2 Mask-RCNN Segmentation Model

For segmentation, we use a ResNet 50 with Feature Pyramid Network (FPN) backbone Mask-RCNN (He et al., 2017), which is a state of the art image segmentation model. We acknowledge that this model is unnecessarily large and complex for a simple segmentation task which could also be solved by non Machine Learning approaches. The benefit of this implementation in our demonstration, however, is that little changes to the general architecture are necessary to adapt it to other datasets or AVR tasks. We do not use a pre-trained checkpoint, as available checkpoints are trained on either ImageNet (Deng et al., 2009) or MSCOCO (Lin et al., 2014), both of which contain real-world images which are very different from the RAVEN panels. We opt for a configuration-stratified dataset split of 90% training data and 5% validation and test data each. We train for 10 epochs, which is a training time of roughly 34 hours using a single NVIDIA GeForce RTX 3090 GPU. Note that the trained model is no general segmentation model alike SAM (Kirillov et al., 2023), but it may still be applicable to other AVR tasks as long as the objects contained in the dataset are simple shapes (e.g. Bongard problems (Bongard, 1970; Nie et al., 2020)).

The evaluation results on the test split are reported in Table 1. For the use within the framework, recall is the most important metric. It is vital that the segmentation model produces shape predictions for every shape in the matrix, as missing shapes may lead to missing solution-relevant information in the discrete problem representation which makes the problem instance unsolvable. Despite the near perfect metrics recorded, the model proved to not perform well enough to be used for full inference within ECo-NSR at closer inspection. This is due to the scores being the average over the IoU thresholds from 0.5 to 0.95 in steps 0.05. High recall at lower IoU thresholds inflates the final average score, despite predictions with the lower IoUs (an exact threshold is hard to quantify) are virtually unusable for our purposes (see Figure 5). Furthermore, we can compute

Table 1: Test results of the trained Mask-RCNN segmentation model. An Intersection over Union (IoU) of $0.5 : 0.95$ means that scores where calculated over the range from 0.5 to 0.95 in steps of 0.05, and then averaged. The value maxDets is the maximum number of instance predictions considered for the score calculation and is only relevant for recall. Note that the low recall for a single prediction results from most panels containing more than one shape, such that perfect recall cannot be achieved.

Metric	IoU	area	maxDets	bounding-boxes	segmentation
Average Precision (AP)	$0.5 : 0.95$	all	100	0.945	0.934
	0.5	all	100	1.000	0.990
	0.75	all	100	0.990	0.990
	$0.5 : 0.95$	small	100	0.915	0.899
	$0.5 : 0.95$	medium	100	0.970	0.962
	$0.5 : 0.95$	large	100	0.976	0.979
Average Recall (AR)	$0.5 : 0.95$	all	1	0.400	0.395
	$0.5 : 0.95$	all	10	0.962	0.949
	$0.5 : 0.95$	all	100	0.962	0.949
	$0.5 : 0.95$	small	100	0.934	0.916
	$0.5 : 0.95$	medium	100	0.985	0.975
	$0.5 : 0.95$	large	100	0.995	0.989

a performance requirement for high-IoU Recall, which shows the disparity of the model achieved and the model required: If the goal is to produce correct mask predictions for a proportion p of the seen matrices, and we assume perfect precision (no false positives) and statistically independent predictions (i.e. every shape is detect with an individual likelihood of p), then the average recall required to achieve this proportion can be calculated as $\sqrt[n\_shapes]{p}$, with n_shapes being the number of shapes in the entire problem. For the most simple case, a RAVEN-F matrix of the `Center` configuration and a relaxed $p = 0.8$, this results in a required recall of $\sqrt[16]{0.8} = 0.986$. For a more strict proportion of 0.95[1] and a fully filled matrix of the `3x3Grid` configuration, the required recall becomes $\sqrt[144]{0.95} = 0.9996$, which is a near perfect model.

4.3 β-TCVAE Representation Model with Mean-Shift Clustering

For representation we train a β-TCVAE (Chen et al., 2018) model with the standard hyperparameters of $\alpha = 1, \beta = 6, \gamma = 1$ and a latent representation with six dimensions. The model is trained on a subset of the generated RAVEN-F dataset which excludes the two configurations `Out-InCenter` and `Out-InGrid`. This is because for these configurations, the outer shapes contain other shapes, which we do not want to be including in our representation. Out of the remaining 50,000 Matrices, we use the bounding box annotations to crop the images to each shape. Crops are then resized to the

1. Given that the representation and reasoning steps introduce additional sources of potential errors, this is not unreasonable.

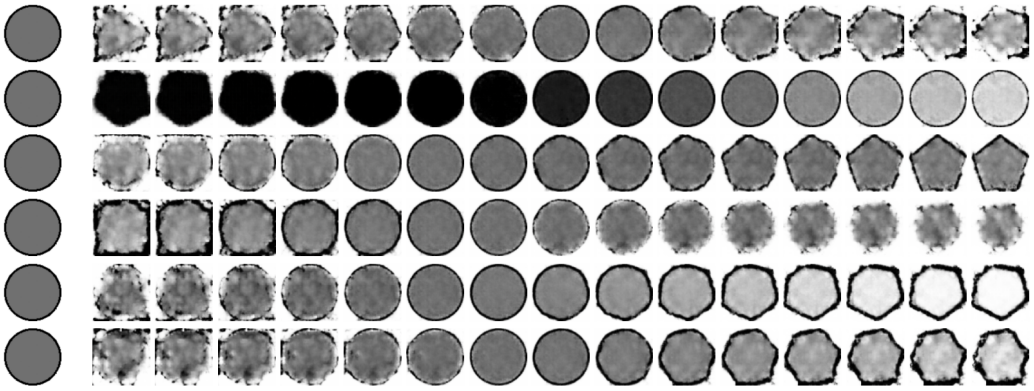
Figure 5: Examples of bounding box and mask predictions of the learned Mask-RCNN segmentation model, with predictions on the left and ground truths on the right respectively. Predictions were filtered to include only predictions with a confidence score of over 0.8.

Figure 6: Example reconstructions generated by our β-TCVAE model, with reconstructions on the left and ground truths on the right respectively. Samples are taken from the test set.

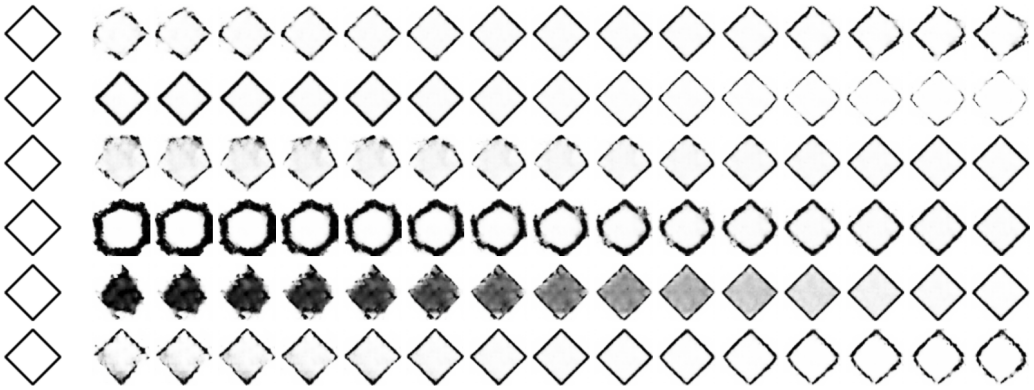
input size of 128x128. The resulting dataset contains a total of 1,916,493 shapes, out of which 80% were used as training data, and 10% each were used as validation and test data. We train the model for 10 epochs, which is a training time of roughly 45 minutes using a single NVIDIA GeForce RTX 3090 GPU.

The training resulted in a mean-squared error of 0.004 for a pixel value range of $0 : 1$, with reconstructions being visually of very high quality (see Figure 6). Still, the model doesn't cleanly disentangle the explanatory features of the data distribution. Figure 7 shows latent traversals for the six latent dimensions learned by our model using three different shapes as basis. We can clearly observe that shape-type and color are dependent on a combination of all six dimensions, indicated by the shapes and colors remaining mostly constant throughout the traversals in the dimensions. While the second dimension in Figures 7a and 7c seem to indicate the encoding of color, the diamond's color in Figure 7b is not affected by that dimension, but only by dimension 5. Finally, there is no indication of any of the dimensions encoding shape rotation; rather, different rotations of the same shape seem to be encoded at entirely different regions in the latent space, indicated e.g. by variations of pentagonal shapes appearing in positive values of dimensions 0 and 3 for the circle basis (see Figure 7a), negative values in dimension 3 for the diamond basis (see Figure 7b), and throughout dimension 5 for the triangle basis (see Figure 7c).

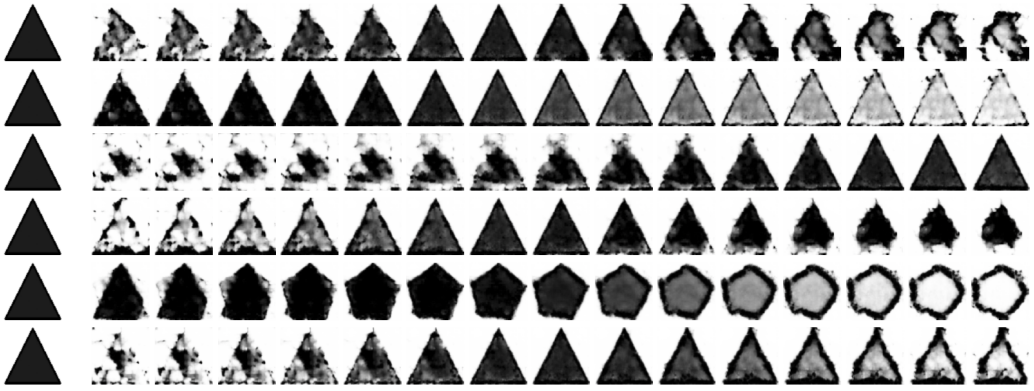
To generate discrete representations from the continuous latent variables produced by our model, we use Mean Shift Clustering (Comaniciu & Meer, 2002), since we perform the clustering on a maximum of 144 samples (which would result from a `3x3Grid` type matrix with 9 shapes in each image) and are thus not affected by the algorithm's poor scalability to large sample sizes. It has the benefit that it does not require the number of clusters to be known beforehand. Exemplary results of such clustering are shown in Figure 8. Despite the model not disentangling well, many clusters show strong similarities among the shapes in them while differentiating themselves from other clusters



(a) Circle-shape representation basis.

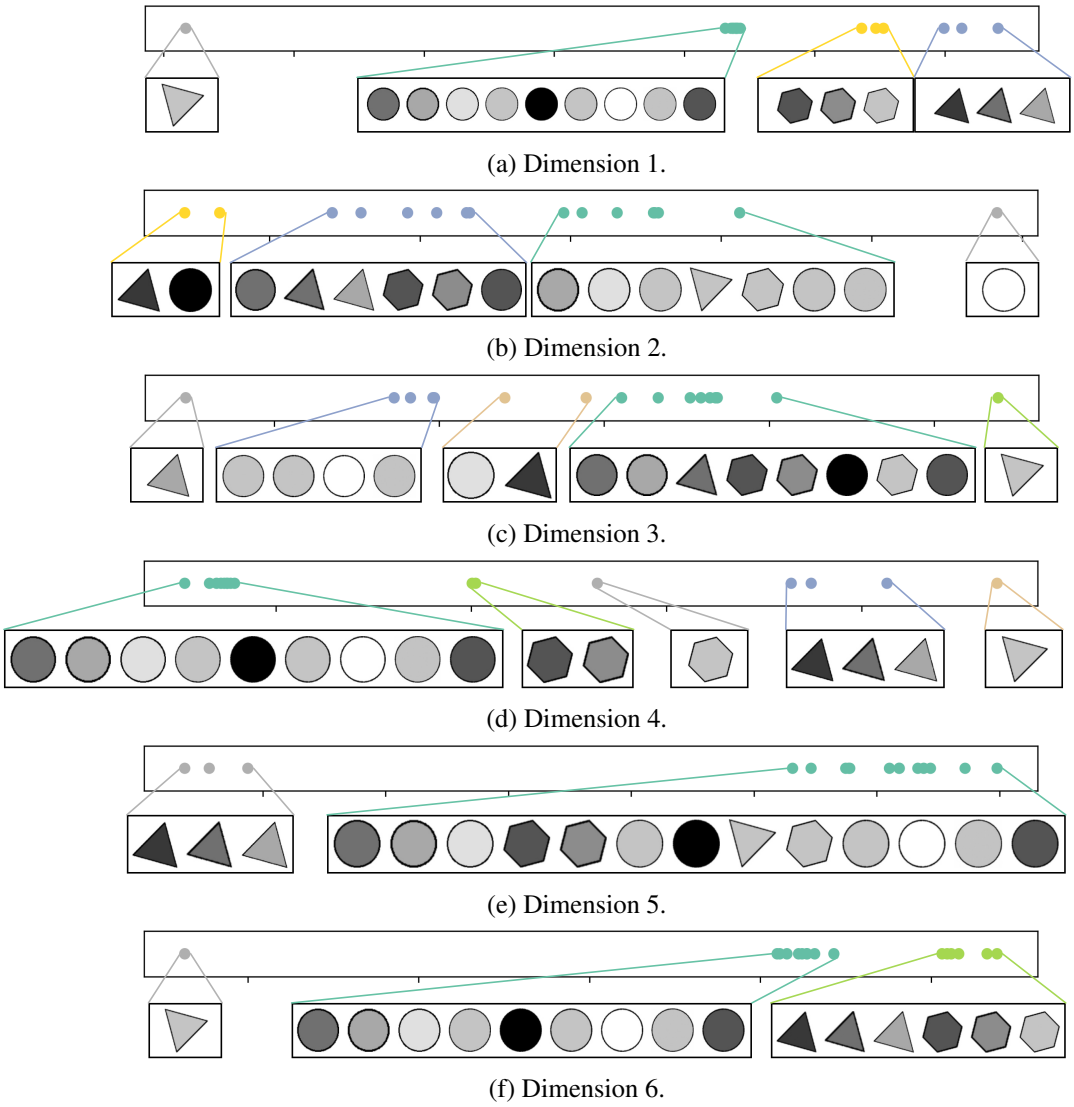


(b) Diamond-shape representation basis.



(c) Triangle-shape representation basis.

Figure 7: Latent traversals for the six dimensions learned by our β-TCVAE model. Each row shows one dimension and varies the latent value at 15 steps in the range of two standard deviations around the mean value for that dimension recorded on the test split.



(a) Dimension 1.

(b) Dimension 2.

(c) Dimension 3.

(d) Dimension 4.

(e) Dimension 5.

(f) Dimension 6.

Figure 8: Dimension-wise clustering results for the shapes of the matrix shown in Figure 1.

for the same dimension. For example, dimension 1 (Figure 8a) perfectly separates shape type, with the triangle oriented differently being an outlier. Dimension 2 (Figure 8b) clusters shape by color, dimension 5 (Figure 8e) recognizes triangles (once again with one outlier). While there are clear redundancies between dimensions, clusters appear to be mostly meaningful and give information about properties of the shapes.

4.4 POPPER as Symbolic Reasoner

We choose POPPER (Cropper & Morel, 2021) as a state of the art ILP system for our symbolic reasoner. We also consider NUMSYNTH (Hocquette & Cropper, 2023b) in our analysis for its ability to learn rules containing constants. To be able to use POPPER, we create a logical structural model of the matrix using structural (which shape belongs to which panel, which panel belongs to which row) and attribute information.

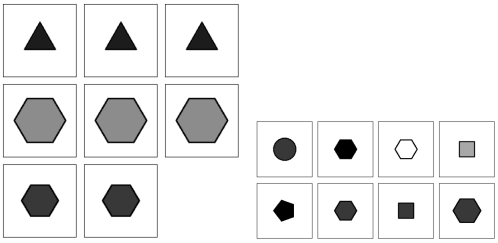
The matrix is modeled hierarchically, in the sense that panels are associated with the shapes they contain using the predicate $has\_shape/2$ with types $has\_shape(panel, shape)$, and shapes are associated with their attributes and attribute values using the predicate $shape\_prop/3$ with types $shape\_prop(shape, attribute, value)$. Additionally, we introduce the predicate $n\_shapes/2$ with types $n\_shapes(panel, value)$, as this is the only attribute in the RAVEN-F dataset associated with the panel itself instead of the individual shapes.

From this, the learning problem is learning the target predicate $row/3$, with each parameter being a panel. The positive examples are the first two rows, so $row(panel_1, panel_2, panel_3)$ and $row(panel_4, panel_5, panel_6)$. It should be noted that it is vital that the learner is constrained to learning only a single rule, as the option to learn multiple rules offers the trivial solution of learning a rule for row 1 and a rule for row 2, not generalizing between the two. Additionally, it is known that *exactly one* of the panels in the answer set makes up the correct answer, so out of the possible examples $\{row(panel_7, panel_8, answer_i)\}_{i=1}^{N}$, N being the size of the answer set, one example must be a positive example, and the others are negative examples. Since the learner does not know which is the correct answer prior to learning the rule, this leaves two possible strategies to incorporate this knowledge to increase the chance of finding the correct rule:

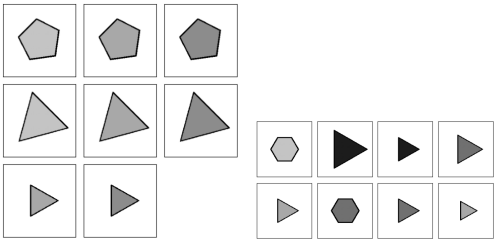
1. *Noisy*: Include the entire answer set as negative examples, and rely on a) the assumption that there exists no complete rule for only the first two rows and b) on Popper's ability to learn from noisy example sets.

2. *Trial-and-Error*: Include any one possible answer as a positive example and the rest as negative examples, learn N programs, and rely on the assumption that there exists only one panel in the answer set, for which a consistent rule can be learned.

Next to the problem information, the background knowledge contains definitions of two types of rules. *General relations* are rules which are required to express some of the rules in common RPM datasets. Importantly, these general rules are not complete formulations of RPM rules, but are necessary components to express the rules. Each relation included in the background knowledge opens the possibility to express even more potential rules. The general relations included are $select/3$ which allows the removal of elements from lists, $sum/3$ allows to add three numerical values, and $add\_1/2$ allows to add 1 to any value. These relations are necessary to express the rule types distribute three, arithmetic and progression in the RAVEN (Zhang et al., 2019) dataset[2]. *Shortcuts* are rules which condense expressions containing many literals and variables into single literals, with the goal of allowing POPPER to learn more complex rules within its computational limits. These shortcuts are $same\_in\_prop/3$ which indicates whether two shapes have the same



(a) Simple matrix containing constant rules in each attribute.



(b) Matrix containing a progression rule in the color attribute.

Figure 9: Visual representations of two example matrices of different difficulties for the POPPER.

value in a given property and saves one literal, and $identical/2$ which indicates whether two shapes are equal in every attribute.

In evaluation, we were able to learn the underlying rule for matrices where the rule for each attribute was constant using the *Trial-and-Error* strategy and representations generated from the learned model. An example of such a simple matrix is shown in Figure 9a. For this matrix, POPPER was able to learn the following rule:

$$row(V0,V1,V2) \leftarrow has\_shape(V1,V4), has\_shape(V0,V3), \\ has\_shape(V2,V5), identical(V5,V4).$$

This is a correct rule for the given matrix. Note that POPPER ignores the shape extracted from the first panel and only uses the shape in the second panel as a comparison for the third panel. In a similar manner, we can formulate the following rule to perfectly cover the matrix shown in Figure 9b:

$$\begin{aligned} row(V0,V1,V2) \leftarrow\ & has\_shape(V1,S0), has\_shape(V2,S1), \\ & shape\_prop(S0,attr5,Val0), shape\_prop(S1,attr5,Val1), \\ & add\_1(Val0,Val1), same\_in\_prop(S0,S1,attr3), \\ & same\_in\_prop(S0,S1,attr4). \end{aligned}$$

Just as the rule learned by POPPER for the simpler matrix, this rule ignores the first panel in the row while maintaining perfect coverage of the problem. The rule shortened in this way contains 7 body literals, 7 variables, and 3 constants. Yet, NUMSYNTH was not able to learn this rule with the maximum body literals set to 8, maximum number of rules set to 1, maximum number of variables set to 8 and maximum number of magic values (i.e. constants) set to 6.

Arithmetic and distribute three rules (Zhang et al., 2019) require even more literals and variables to be formulated, as the first panel cannot be ignored in this case. Additionally, note that for the

2. The RAVEN-F (Benny et al., 2021) dataset contains the same rule types.

matrix in Figure 9b, the rules governing the shape and size attributes were still constant. Matrices requiring more than one complex rule, let alone contain more than a single shape per image, result in even larger rule formulations.

5. Discussion

Our example implementation of the ECo-NSR framework demonstrates that segmentation, representation and reasoning can be combined into a unified system. While we so far are not able to solve nontrivial matrix problems end-to-end due to the insufficient performance of our segmentation model and reasoning approach, we are able to learn a representation model which uses the modality of the segmentation output to represent individual shapes instead of entire panels. We are also able to create a discrete structural representation of matrix problems from the continuous representations produced by our model which we can use as input to an ILP system.

Most notably, while the structural model would need to be adapted, our framework isn't constrained to only RPM problems but can be applied to virtually any AVR problem which relies on relationships between shapes or their attributes and offers a dataset annotated with segmentation masks or at least bounding boxes. Due to the framework's modularity, not even that is necessary as long as the objects are simple enough that they can be detected using classical computer vision approaches such as Scale-Invariant Feature Transform (Lowe, 1999) or the Canny Edge Detector (Canny, 1986).

To improve the segmentation model over our baseline, ensemble methods could be considered. However, with the large size, training, and inference cost of these models, sustainability must also be taken into account. This is mostly an engineering problem which can solved with simple albeit expensive hyperparameter search. With the introduction of foundation models for image segmentation comes the option of using e.g. SAM (Kirillov et al., 2023) to detect objects in a zero-shot manner, removing the need for training a dataset-specific segmentation model. Such mask predictions can be combined with representations from a foundation model such as the recent DINOv3 (Siméoni et al., 2025), removing the need for dataset-specific training completely and paving the way towards a general AVR solver.

A possible line of future work is adding processing to mitigate noisy segmentation and representation. As touched upon in Section 4.2, the symbolic reasoner relies on accurate feature representations, which in turn rely on accurate segmentation. Noise is a very complex problem to solve, as it would need to be handled differently depending on its source. In the present example, noise in the representation may come from i) shapes which are present in the image but were not recognized by the segmentation model, ii) shapes which are *not* present in the image but were incorrectly recognized by the segmentation model, and iii) shapes which were correctly recognized but represented inconsistently with other shapes by the representation model. Further, dealing with noise requires knowledge of the underlying analogy problem. For example, in RPM problems, it makes a difference whether noise is present in one of the analogy bases, the target row, or the answer options. Implementing compensations specifically for this won't transfer to other AVR problems. As of right now, this is an unsolved problem.

The most impactful limitation of this work is the choice of the reasoner, or rather the match of the reasoner and our chosen structural model. While POPPER achieves impressive performance on many learning tasks (Cropper & Morel, 2021), our structural model of RPMs results in rules requiring many literals and variables, which increase POPPER's search space to a point where the system can't find programs in reasonable time (under an hour). We suggest future work in this direction to investigate different structural models and/or learning systems. Aleph (Srinivasan, 2001) uses other learning strategies as POPPER and is not as constrained by the size of the learned rules. In addition, Aleph natively supports constants. Further, it could be considered to substitute the ILP learner using a different paradigm entirely. Anti-unification (Cerna & Kutsia, 2023) is a powerful approach for analogical reasoning and may be able to support or replace the ILP learner altogether.

6. Code Availability

Code for the exemplary model described in Section 4 including base background knowledge and bias for POPPER is available at `https://github.com/johannes-langer/ECo-NSR/`.

References

Bengio, Y., Courville, A. C., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, *35*, 1798–1828.

Benny, Y., Pekar, N., & Wolf, L. (2021). Scale-localized abstract reasoning. *CVPR 2021* (pp. 12557–12565).

Bongard, M. M. (1970). *Pattern recognition*. Rochelle Park, NJ: Spartan Books.

Canny, J. F. (1986). A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, *8*, 679–698.

Carpenter, P. A., Just, M. A., & Shell, P. (1990). What one intelligence test measures: a theoretical account of the processing in the Raven Progressive Matrices Test. *Psychological Review*, *97*, 404–431.

Cerna, D. M., & Kutsia, T. (2023). Anti-unification and generalization: A survey. *IJCAI 2023* (pp. 6563–6573).

Chen, T. Q., Li, X., Grosse, R. B., & Duvenaud, D. (2018). Isolating sources of disentanglement in variational autoencoders. *NeurIPS 2018* (pp. 2615–2625).

Chollet, F. (2019). On the measure of intelligence. *CoRR*, *abs/1911.01547*.

Chollet, F., Knoop, M., Landers, B., Kamradt, G., Jud, H., Reade, W., & Howard, A. (2024). ARC Prize 2024.

Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, *24*, 603–619.

Cropper, A., & Dumancic, S. (2022). Inductive logic programming at 30: A new introduction. *J. Artif. Intell. Res.*, *74*, 765–850.

Cropper, A., & Morel, R. (2021). Learning programs by learning from failures. *Mach. Learn.*, *110*, 801–856.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. *CVPR09*.

Evans, T. G. (1964). A heuristic program to solve geometric-analogy problems. *AFIPS 1964 (Spring)* (pp. 327–338).

Falkenhainer, B., Forbus, K. D., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence*, *41*, 1–63.

Flener, P., & Schmid, U. (2008). An introduction to inductive programming. *Artif. Intell. Rev.*, *29*, 45–62.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, *7*, 155–170.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. B. (2017). Mask R-CNN. *ICCV 2017* (pp. 2980–2988).

Hernández-Orallo, J., Martínez-Plumed, F., Schmid, U., Siebers, M., & Dowe, D. L. (2016). Computer models solving intelligence test problems: Progress and implications. *Artif. Intell.*, *230*, 74–107.

Higgins, I., Matthey, L., Pal, A., Burgess, C. P., Glorot, X., Botvinick, M. M., Mohamed, S., & Lerchner, A. (2017). beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR 2017*.

Hocquette, C., & Cropper, A. (2023a). Learning programs with magic values. *Mach. Learn.*, *112*, 1551–1595.

Hocquette, C., & Cropper, A. (2023b). Relational program synthesis with numerical reasoning. *AAAI 2023* (pp. 6425–6433).

Hocquette, C., Langer, J., Cropper, A., & Schmid, U. (2024). Can humans teach machines to code? *CoRR*, *abs/2404.19397*.

Hofstadter, D., & Group, F. A. R. (1995). *Fluid concepts and creative analogies: Computer models of the fundamental mechanisms of thought*. New York: Basic Books.

Holyoak, K. J. (2012). Analogy and relational reasoning. In *The oxford handbook of thinking and reasoning*, 234–259. New York: Oxford University Press.

Hummel, J. E., & Holyoak, K. J. (2019). LISA: A computational model of analogical inference and schema induction. *CogSci 2019* (pp. 352–357).

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. *ICLR 2014*.

Kirillov, A., et al. (2023). Segment anything. *ICCV 2023* (pp. 3992–4003).

Kunda, M., McGreggor, K., & Goel, A. K. (2013). A computational model for solving problems from the Raven's Progressive Matricesintelligence test using iconic visual representations. *Cognitive Systems Research*, *22-23*, 47–66.

Lin, T., Maire, M., Belongie, S. J., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. *ECCV 2014* (pp. 740–755).

Lovett, A., & Forbus, K. (2017). Modeling visual problem solving as analogical reasoning. *Psychological Review*, *124*, 60–90.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. *ICCV 1999* (pp. 1150–1157).

Malkinski, M., & Mandziuk, J. (2022). Deep learning methods for abstract visual reasoning: A survey on Raven's Progressive Matrices. *CoRR*, *abs/2201.12382*.

Malkinski, M., & Mandziuk, J. (2023). A review of emerging research directions in abstract visual reasoning. *Inf. Fusion*, *91*, 713–736.

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., & Terzopoulos, D. (2022). Image segmentation using deep learning: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, *44*, 3523–3542.

Mitchell, M. (1993). *Analogy-making as perception: A computer model*. MIT Press.

Muggleton, S. (1991). Inductive logic programming. *New generation computing*, *8*, 295–318.

Nie, W., Yu, Z., Mao, L., Patel, A. B., Zhu, Y., & Anandkumar, A. (2020). Bongard-logo: A new benchmark for human-level concept learning and reasoning. *NeurIPS 2020*.

van den Oord, A., Vinyals, O., & Kavukcuoglu, K. (2017). Neural discrete representation learning. *NeurIPS 2017* (pp. 6306–6315).

Raven, J. C. (2000). The Raven's Progressive Matrices: Change and stability over culture and time. *Cognitive psychology*, *41*, 1–48.

Raven, J. C., & Court, J. H. (1938). *Raven's progressive matrices*.

Ravi, N., et al. (2025). SAM 2: Segment anything in images and videos. *ICLR 2025*.

Santoro, A., Hill, F., Barrett, D. G. T., Morcos, A. S., & Lillicrap, T. P. (2018). Measuring abstract reasoning in neural networks. *ICML 2018* (pp. 4477–4486).

Shah, V., Sharma, A., Shroff, G., Vig, L., Dash, T., & Srinivasan, A. (2022). Knowledge-based analogical reasoning in neuro-symbolic latent spaces. *IJCLR 2022* (pp. 142–154).

Siméoni, O., et al. (2025). Dinov3. *arXiv preprint arXiv:2508.10104*.

Srinivasan, A. (2001). The aleph manual.

Takida, Y., et al. (2022). SQ-VAE: variational bayes on discrete representation with self-annealed stochastic quantization. *ICML 2022* (pp. 20987–21012).

Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological Review*, *55*, 189–208.

Velickovic, P., & Blundell, C. (2021). Neural algorithmic reasoning. *Patterns*, *2*, 100273.

Wang, X., Chen, H., Tang, S., Wu, Z., & Zhu, W. (2022). Disentangled representation learning. *arXiv preprint arXiv:2211.11695*.

Weller, S., & Schmid, U. (2006). Solving proportional analogies by E–generalization. *29th Annual German Conference on Artificial Intelligence* (pp. 64–75). Springer.

Wiese, E., Konderding, U., & Schmid, U. (2008). Mapping and inference in analogical problem solving – as much as needed or as much as possible? *CogSci 2008* (pp. 927–932).

Wu, Y., Dong, H., Grosse, R. B., & Ba, J. (2020). The scattering compositional learner: Discovering objects, attributes, relationships in analogical reasoning. *CoRR*, *abs/2007.04212*.

Zhang, C., Gao, F., Jia, B., Zhu, Y., & Zhu, S. (2019). RAVEN: A dataset for relational and analogical visual reasoning. *CVPR 2019* (pp. 5317–5327).

Zhao, S., You, H., Zhang, R., Si, B., Zhen, Z., Wan, X., & Wang, D. (2023). An interpretable neuro-symbolic model for Raven's Progressive Matrices reasoning. *Cogn. Comput.*, *15*, 1703–1724.