Thinking in Images: Analysing Concept Learning in Neural Networks by Generating Bongard Problems

OLLSCOIL NA GAILLIMHE
UNIVERSITY OF GALWAY

Pavel Antonov
School of Computer Science
University of Galway

Supervisor(s)
Dr. James McDermott

In partial fulfillment of the requirements for the degree of
MSc in Computer Science (Artificial Intelligence - Online)
31st August 2025

DECLARATION I, Pavel Antonov, hereby declare that this thesis, titled "Thinking in Images: Analysing Concept Learning in Neural Networks by Generating Bongard Problems", and the work presented in it are entirely my own except where explicitly stated otherwise in the text, and that this work has not been previously submitted, in part or whole, to any university or institution for any degree, diploma, or other qualification.

Signature: _____

Abstract

How do neural networks learn and represent visual concepts? This paper introduces a novel approach using image generation to analyse concept learning. A simple text-to-image transformer architecture is trained using only perceptual loss and generates visual representations of concepts from existing Bongard Problems using both symbolic and natural language. By leveraging contrastive pairs from all 100 Bongard Problems, the traditionally data-scarce domain is made more tractable for experimentation. Results showed variation in concept learning for this architecture: some concepts such as relative size and unusual shape structure were generated with remarkable clarity, while others like precise counting and spatial relationships remained fuzzy or failed to form. This disparity emerges despite the minimal architecture, suggesting that certain visual concepts are more naturally learned through perceptual similarity alone.

Keywords: Neural Networks, Concept Learning, Image Generation, Bongard Problems, Symbolic Language, Perceptual Loss

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Understanding how neural networks learn and represent visual concepts remains one of the fundamental challenges in Artificial Intelligence. While neural networks achieve remarkable performance on image classification, object detection, and visual recognition tasks, this thesis introduces a novel approach to investigating concept learning. Image generation is used to probe how neural networks examine visual concepts when trained exclusively through Perceptual Loss. Mikhail M. Bongard designed Bongard Problems, a set of 100 visual puzzles requiring abstract reasoning and visual understanding to solve [2]. While the Artificial Intelligence landscape was quite different in the 1960s when these problems were developed, Bongard Problems proved that visual systems were no where close to human ability [3]. The same is still true today as even state-of-the-art models like o1, GPT-4o, Claude 3.5 and Gemini 2.0 struggle with solving Bongard Problems [4].

Instead of solving Bongard Problems, this thesis probes concept understanding using a simple neural network trained on perceptual similarity. The model in this thesis demonstrates an ability to output visual representations that resemble what cognitive scientists call prototype concepts [5]. Rather than generating per-

fect reproductions, the model learns to create prototypical representations that embody the core visual features distinguishing concepts.

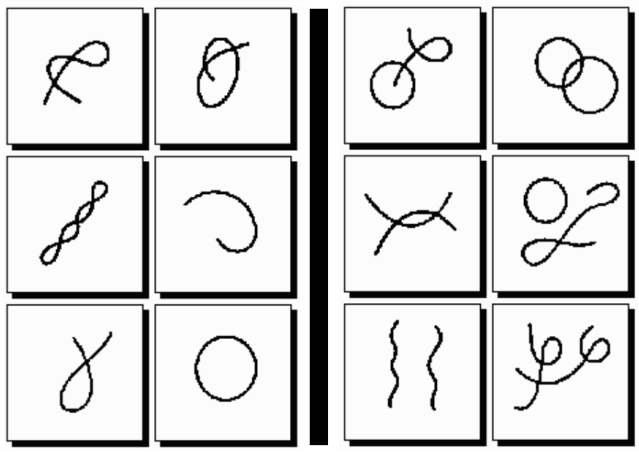

Figure 1.1: Showcasing Bongard Problem#31. Here the solver is presented with two groups of images, with six distinct images in each one. The solver must then deduce a rule that separates these two groups. The solution is "one line vs two lines". Images from foundalis.com [1] by Mikhail M. Bongard [2]

1.1 Motivation

Bongard Problems present a unique challenge to Artificial Intelligence due to their defining characteristics. Data is limited, and solving them requires multimodal capabilities to convert visual elements into language that explains the underlying rule. The issue of data scarcity, in particular, motivated the development of a generative approach in which language descriptions could be used to produce new Bongard Problems. This would expand the dataset and, in turn, enable models to create visual feedback loops to verify their solutions. The idea to create a generative model was inspired by Zhang et al's [6] work where they used a Convolutional Neural Network (CNN) to solve Raven Progressive Matrices (RPMs) [7, 8]. Zhang et al's CNN would analyse visual features, and then according to

hard-coded rules, convert visual features into probabilistic values for a fixed set of attributes. Then according to the attributes which had the highest probabilistic values, an answer would be derived. There is a key sentence in Zhang et al's paper that prompted the idea for a generative model:

> "Furthermore, we show that probabilistic scene representation learned by the PrAE learner can be used to generate an answer when equipped with a rendering engine."[6]

However, rendering an image based on probabilistic values was only used for transparency and for readability. This rendered image was not used to improve the model's ability to solve RPMs. Including both points, this identified a gap that could be addressed by a text-to-image generator.

While developing the text-to-image transformer to create more Bongard Problems, the model revealed interesting behaviour as it was generating visual representations of concepts. This discovery changed the focus of the thesis to research the neural network's ability to learn and represent visual concepts through generation. After all, the architecture is simple, yet the model unexpectedly learned to capture abstract visual concepts.

1.2 Research Questions

The following research questions will be answered in the thesis:

RQ 1 - How do neural networks learn visual concepts when trained exclusively with perceptual loss?

RQ 2 - Which types of visual concepts are more naturally learnable through perceptual similarity alone?

RQ 3 - How does linguistic representation affect visual concept learning in generative models?

1.3 Thesis Structure

Chapter 1 - The Introduction provides a brief overview and the motivation for this research.

Chapter 2 - Background will present information about the topic and related work.

Chapter 3 - Methodology will describe the datasets, the model architecture, its development, and the design decisions that led to the final version.

Chapter 4 - Experiments will in detail explain how the tests conducted and how the results were evaluated.

Chapter 5 - Results will present the outcomes of the experiments and analyse them in relation to the research questions.

Chapter 6 - Conclusion will evaluate the thesis, discuss any limitations, contributions and future work.

Chapter 2

Background

2.1 Neural Networks

Neural networks are computational models inspired by the structure and function of biological neurons in the brain. At their core, they consist of interconnected nodes (artificial neurons) organised in layers that process and transform input data to produce desired outputs.

A typical neural network comprises of an input layer, one or more hidden layers, and an output layer. Each neuron receives weighted inputs from connected neurons and passes transformed information forward through the network. The connections between neurons have adjustable weights that determine the strength of information flow. Neural networks learn by adjusting these connection weights through a process called backpropagation. During training, the network processes input data, compares its predictions to the desired outputs according to a training signal and then updates the weights based on the loss. This iterative process allows the network to gradually improve its performance on the given task.

2.2 Pixel reconstruction losses

L1 and L2 losses are often preferred to as pixel reconstruction loss in image generation tasks. This is because they each examine the difference in pixels for a given input and target input. The main difference between the L1 and L2 is how they weigh these differences.

2.2.1 L1 loss

L1 loss, also known as Mean Absolute Error (MAE), measures the average absolute difference between predicted and target values. L1 loss is computed as:

$$L_1 = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

For image generation, y_i represents the target pixel values and $\hat{y}_i$ the generated pixel values. The linear penalty structure of L1 loss creates a constant gradient for all differences, regardless of their size. This ensures outliers do not contribute disproportionally to the total loss.

2.2.2 L2 loss

L2 Loss or also known as Mean Square Error (MSE), calculates the squared difference between the predicted and target values. L2 loss is computed as:

$$L_2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

For image generation, y_i represents the target pixel values and $\hat{y}_i$ the generated pixel values. The quadratic penalty structure of L2 loss creates large gradients for big differences. This ensures large errors contribute disproportionately more to the total loss than smaller errors.

2.2.3 L1 and L2 efficacy

Both pixel reconstruction losses have demonstrated effectiveness in image generation tasks. In particular, L2 has been successfully used as a training signal to improve image quality in high resolution image generation tasks [9]. Whereas L1 has proven effective in image to image translation tasks, such as semantic segmentation. L1 as a training signal points the model to outputs that closely match the ground truth while ensuring the image does not become blurry [10].

2.3 Convolution Neural Networks

Convolutional Neural Networks (CNNs) are specialised neural network architectures designed to process grid like data, particularly images. Images are processed through convolution, where small matrices of weights are scanned across an image to detect patterns like edges or textures. Pooling layers are also applied where the image is downsized while maintaining either the strongest features, or an average of the features. This creates a neural structure where deeper layers can detect more complex patterns by combining the simpler patterns found by earlier layers. This lets CNNs recognise deep patterns which capture important structure in images.

2.3.1 VGG

VGGs are a type of convolutional neural network designed by Simonyan and Zisserman in 2014 [11]. VGG can be described as very deep convolutional neural networks. The architecture consists of blocks of multiple convolutional layers with small matrices followed by a pooling layer which takes the strongest features. This allows VGG to capture increasingly complex patterns as the image is downsized. VGG was able to demonstrate significant improvements in image classification

accuracy in ImageNet compared to previous approaches. ImageNet is a large dataset consisting of 14 million real life images of various objects, animals and scenes [12]. The learned features from VGG are also highly transferable, making pretrained VGG models effective for various computer vision tasks.

2.3.2 Perceptual Loss

Perceptual loss is a training signal which was first introduced by Johnson et al. in 2016 as an alternative to pixel reconstruction losses to train neural networks for image tasks [13]. Pixel reconstruction losses do not always correspond to perceptual differences perceived by humans [14]. Two images might look similar while containing very large pixel differences. Unlike L1 and L2 which focus on pixel differences, perceptual loss measures differences between feature representations which are extracted from pretrained CNNs. This encourages models to generate images to have similar characteristics as the target image. Typically VGG which has been pretrained on ImageNet is used for perceptual loss.

2.4 CLIP

Contrastive Language-Image Pre-training (CLIP) is a multi-modal neural network developed by Radford et al. at OpenAI in 2021 [15]. CLIP learns visual concepts with labelled image data. The training dataset consists of 400 million images containing meta data as labels which have been scraped from the internet. CLIP models contain two encoders in their architecture, one for images and one for text. These encoders process the respective medium and generate representations. CLIP then learns to associate images with their corresponding text by maximising the similarity between correct image text pairs and minimising incorrect image text pairs. This creates a rich semantic embedding where text is

mapped to an image space and vice versa. CLIP is highly transferable, making it a great model for improving image retrieval tasks and models which generate images from text descriptions.

2.4.1 CLIP Loss

CLIP loss leverages the semantic embeddings from pretrained CLIP models as a training signal for image generation tasks. Models like ViT-B/32, ViT-B/16, and ViT-L/14 are all available to use and come pretrained from the CLIP python library. CLIP loss is typically computed by encoding both the generated image and target text description and measuring the similarity between both. This encourages a model trained with CLIP loss to generate images which have similar semantic content to the target text. CLIP loss has been shown to improve the semantic alignment between generated images and text descriptions [16].

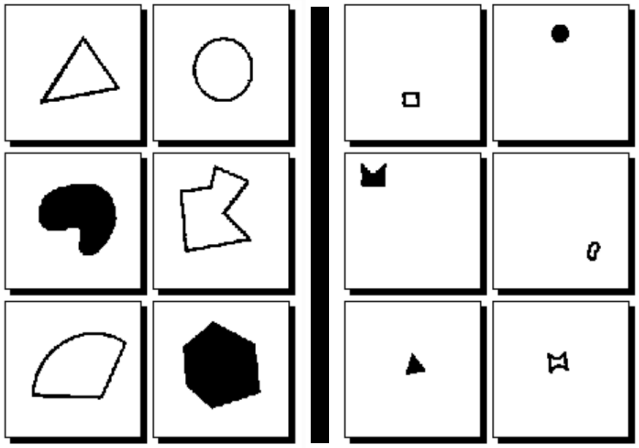

Figure 2.1: BP#2 as an example for why all images are necessary for the concept "Big vs Small". Images from foundalis.com [1] by Mikhail M. Bongard [2]

2.5 Bongard Problems

Bongard Problems have proven to be a great domain to try novel strategies for solving problems where data is limited and highly varied [3, 17, 18, 19, 20], while also inspiring researchers to create new datasets[21, 22, 23] to test AI systems in other creative ways.

Bongard Problems consist of twelve images arranged in two groups of six, where the task is to identify the underlying concept that distinguishes the left group from the right group. A key characteristic of Bongard Problems is that solving them requires analysing both sides to understand the full context of the concept. In most cases, every image plays a crucial role in ensuring the concept is clearly defined. For example, in BP#2 shown in Figure 2.1, without certain images, the concept "Big vs Small" could be misinterpreted as "Big solid figures vs small coloured figures".

2.5.1 Contrastive Image Pairs

However, Youssef et al. has shown that contrastive image pairs from opposing groups retain sufficient information about the underlying concept even when separated from their original context [24]. While all twelve images together provide complete definitional clarity, individual pairs from opposite sides still embody the same conceptual distinction. This property suggests that opposing examples can effectively capture the essence of visual concepts without requiring the full problem context.

2.5.2 No Objects Exist Without Context

Linhares presents a fundamental philosophical argument about the nature of Bongard Problems, asserting that Bongard Problems contain no fixed objects, only descriptions that are valid for specific geometric arrangements [25]. This perspective suggests that any linguistic system can be used to describe these arrangements, provided it accurately captures the underlying geometric relationships.

Although, the validity of concept descriptions is inherently subjective and context dependent. What constitutes a "correct" description depends on the conceptual framework of the interpreting community. For instance, descriptions that are meaningful within one linguistic or cultural context may appear incomprehensible to another, yet both remain to be equally valid representations of the same geometric arrangements. This subjectivity suggests that even arbitrary labels could theoretically capture the essence of a visual concept.

2.5.3 Symbolic Language

Depeweg et al. developed a symbolic language that successfully described 39 out of 100 original Bongard Problems [18, 19]. Depeweg et al. designed a system which uses hard-coded visual processing functions to detect shapes, properties and relationships which are then converted into this symbolic language. Their system successfully solved 35 out of 39 problems by correctly mapping visual features into symbolic language which correctly described the underlying rule. While very impressive compared to previous approaches [3], the symbolic language is limited to what the system is able to map.

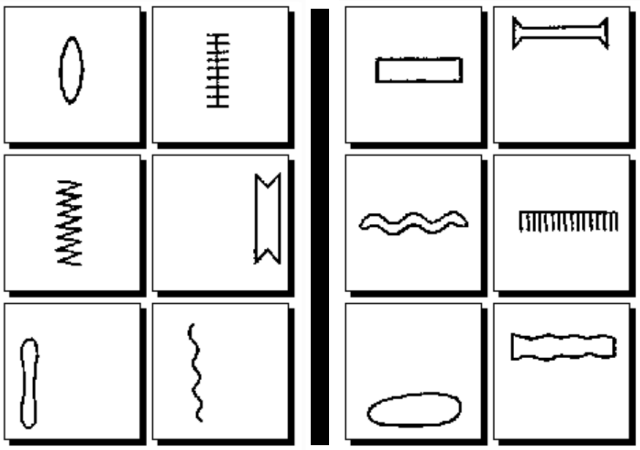

Figure 2.2: Example of the symbolic language applied to BP#7. The problem can be described as `LEFT(GREATER(FIGURES,ORIENTATION))` or equivalently `RIGHT(LESSER(FIGURES,ORIENTATION))`, where low orientation corresponds to horizontally oriented figures and high orientation to vertically oriented figures. The solution is "taller than wide vs. wider than tall." Images from foundalis.com [1], originally by Mikhail M. Bongard [2].

The symbolic language uses expressions like `"LEFT(EXISTS(TRIANGLES))"` to indicate that triangles exist on the left side of a Bongard Problem but do not exist on the right side. A key characteristic of this symbolic language approach is its specification of particular sides when describing concepts. This directional specification means that a single sentence describes only one side of the problem,

with the opposing side being implied to be a negation or contrast depending on the sentence vocabulary. For example, a description like `"LEFT(GREATER(FIGURES,ORIENTATION))"` indicates that the left side contains figures with greater orientation, while implying the right side has figures with lesser orientation. Overall, this symbolic language offers a structured framework for describing the visual concepts found in Bongard Problems.

Chapter 3

Methodology

This chapter details the iterative development process of a text-to-image transformer that began as an attempt to generate additional Bongard Problem data but evolved into an investigation of visual concept learning. The following section outlines how data was created from scratch, the architectural evolution, loss function discoveries, and training methodologies that led to visual representations of the underlying concepts. Throughout development, model improvements were evaluated through visual assessment of generated image quality. This subjective evaluation focused on whether outputs exhibited clear geometric arrangements with minimal noise, resembling Bongard Problem concepts.

3.1 Model Architecture

First, text needs to be converted to a tokenised sequence for the model. This is done by a simple custom tokeniser which converts text into discrete tokens and builds a vocabulary of the entire training set. The model then processes tokenised text sequences and generates corresponding paired image outputs through a dual generator system. The architecture begins with an embedding layer that

combines learned token embeddings with positional encodings. These combined embeddings are then processed by a TransformerEncoder (PyTorch) built from 8 TransformerEncoderLayers (PyTorch) with an embedding dimension size of 512 and 16 attention heads. The transformer encoder output is then aggregated using attention-weighted pooling to create a context vector for image generation. This context vector is projected through a linear layer and then split into two embeddings that feed the left and right image generators respectively.

Each image generator architecture begins with a linear projection that doubles the embedding dimensionality to 1024, followed by ReLU activation and dropout (0.2) for regularization. The embedding is then projected to 512×8×8 feature maps and processed through a series of transposed convolutional layers that progressively upsample to 128×128 resolution: 512→256→128→64→32→1 channels. Each upsampling layer uses 4×4 kernels with stride 2 and padding 1, followed by BatchNorm2d, ReLU activation, and Dropout2d (0.1) regularization. The final layer employs a 3×3 convolution with Tanh activation to produce greyscale images in the [-1, 1] range. Both generators share identical architecture.

3.2 Data

To train the model, rows of data was created from scratch in the format:

- **Input:** text → **Output:** left image, right image

From the first 100 Bongard Problems, only 1,200 images are available. By leveraging the model architecture and contrastive image pairs from Section 2.5.1, the dataset expands significantly. Each Bongard Problem now results in 36 unique pairs of opposing images (6×6 combinations), giving 3,600 rows of base data. The image data consists of high quality Bongard Problem images of size 432×432 which were made available by Depeweg et at [18, 19].

The data chapter will describe three datasets which were created from scratch specifically for this model and to answer the research questions:

- **Symbolic Dataset** – 8,532 rows of data using the symbolic language to describe concepts.
- **English Dataset** – 3,600 rows of data using natural language to describe of concepts.
- **Minimal Dataset** – 3,600 rows of data using simple numerical labels to describe concepts.

3.2.1 Symbolic Language as Concept Specification

The Symbolic Dataset uses Depeweg et al's symbolic language as described in section 2.5.3. Without the limitation of a visual processing system, words were added to the vocabulary which expanded the coverage from 39 to 75 Bongard Problems. Certain Bongard Problems were difficult to describe in the symbolic language and were excluded from the dataset. To address RQ3, the same problems are described multiple times using slightly different vocabulary but maintaining the same meaning. This builds up a larger dataset to establish semantic understanding between the symbolic language and visual features. As an example, BP#2, which represents the concept "Big vs Small", was described eight different ways in the Symbolic Dataset, providing $8 \times 36 = 288$ rows of data:

1. `LEFT(EXISTS(BIG(FIGURES)))` $\rightarrow$ BP#2
2. `LEFT(EXISTS(HIGH(FIGURES,SIZE)))` $\rightarrow$ BP#2
3. `LEFT(EXACTLY(1,BIG(FIGURES)))` $\rightarrow$ BP#2
4. `LEFT(EXACTLY(1,HIGH(FIGURES,SIZE)))` $\rightarrow$ BP#2

5. `RIGHT(EXISTS(SMALL(FIGURES)))` → BP#2
6. `RIGHT(EXISTS(LOW(FIGURES,SIZE)))` → BP#2
7. `RIGHT(EXACTLY(1,SMALL(FIGURES)))` → BP#2
8. `RIGHT(EXACTLY(1,LOW(FIGURES,SIZE)))` → BP#2

The Symbolic Dataset was finalised with 8,532 rows of data and 71 words in the vocabulary. The number of representations per problem varied with the complexity of the geometric arrangement. On average, each Bongard Problem had four representations, with the total representations per problem ranging from one to eight. However, manually constructing these sentences was time consuming, difficult, and could not cover all 100 problems. This motivated the exploration of alternative linguistic approaches to describe the concepts.

3.2.2 English Language as Concept Specification

To address RQ3, the English Dataset was designed as a direct comparison between many vs one linguistic representation per Bongard Problem. Concepts were described using natural language, with solution answers from OEBP.org [26] covering all 100 problems. Each problem was given a single representation, resulting in $100 \times 36 = 3{,}600$ rows of data.

1. "Empty image vs non-empty image" → BP#1
2. "Big vs small" → BP#2
3. "Hollow outline vs filled in solid" → BP#3

The English Dataset was finalised with 3,600 rows of data describing each Bongard Problem once with an English sentence.

3.2.3 Minimal Language as Concept Specification

The Minimal Dataset uses simple numerical labels "BP1", "BP2", "BP3", etc to describe Bongard Problems. For all training images from BP#1, the input text will be "BP1" and for all training images in BP#2, the input text will be "BP2", and so on up to BP#100. The Minimal Dataset contains no descriptions of the concept, eliminating semantic meaning and making the model rely solely on the visual contents of the training data. To address RQ3, this dataset serves as a direct contrast by testing concept learning in the complete absence of linguistic representation.

1. "BP1" $\rightarrow$ BP#1
2. "BP2" $\rightarrow$ BP#2
3. "BP3" $\rightarrow$ BP#3

The Minimal Dataset was finalised with 3,600 rows of data describing each Bongard Problem once with a numbered label.

3.3 Pixel Reconstruction Loss

The first loss functions used were L1 and L2 pixel reconstruction losses. During training, the loss was computed by minimising the difference between each generated image and a valid reference image from the corresponding Bongard Problem. Unlike image to image tasks where a unique ground truth image exists, Bongard Problems allow multiple valid outputs for the same input description. Due to the network being deterministic, it cannot produce this diversity of solutions. Instead, pixel reconstruction losses encourage convergence toward the pixel average of all valid outputs, resulting in noisy clusters of black pixels rather than coherent geometric arrangements. However, examination of these outputs

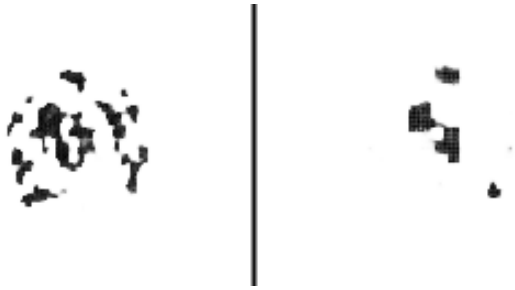

Figure 3.1: Model outputs trained with pixel reconstruction loss for the input `"LEFT(EXISTS(BIG(FIGURES)))"`. Although the images are noisy and lack coherent shapes, they still reflect the underlying rule of BP#2 "Big vs Small", as larger figures consistently contain more black pixels than smaller ones.

revealed adherence to underlying conceptual rules. Simple Bongard Problems like BP#2 shown in figure 3.1 comply with the underlying rule of Big vs Small, as larger shapes contain more black pixels than smaller shapes when averaged across multiple examples. While pixel reconstruction loss appeared to create representations of simple concepts, the outputs lacked sufficient clarity for rigorous analysis of concept learning. This averaging effect is similar to mode collapse in generative models and prompted exploration for more sophisticated loss functions to produce coherent geometric arrangements.

3.4 Perceptual Loss and CLIP Loss

The limitations of pixel reconstruction loss led to the adoption of Perceptual loss and CLIP loss for training. Perceptual loss was implemented using a pretrained VGG19 model from the torchvision python library. In the initial approach, feature representations were extracted from multiple convolutional blocks of VGG19, ranging from early layers that detect edges to later layers that capture more textural detail. For each block, the L2 distance was calculated between the feature maps of the generated image and a reference training image, and these values were summed to form the total perceptual loss. The following VGG19 feature blocks

were extracted:

- Layers 0-2: Conv2d → ReLU → **Conv2d (Layer 2)**
- Layers 3-7: ReLU → MaxPool2d → Conv2d → ReLU → **Conv2d (Layer 7)**
- Layers 8-12: ReLU → MaxPool2d → Conv2d → ReLU → **Conv2d (Layer 12)**
- Layers 13-21: ReLU → Conv2d → ReLU → Conv2d → ReLU → MaxPool2d → Conv2d → ReLU → **Conv2d (Layer 21)**
- Layers 22-30: ReLU → Conv2d → ReLU → Conv2d → ReLU → MaxPool2d → Conv2d → ReLU → **Conv2d (Layer 30)**

Both convolutional outputs and ReLU activations were tested with repeated runs. ReLU activations consistently led the model to converge to white backgrounds for all inputs, whereas convolutional outputs produced clear geometric arrangements.

For implementing CLIP loss, the ViT-B/32 model from the CLIP Python library was used [27]. ViT-B/32 encodes both text and images into embeddings. During training, generated images and non-tokenised text were input into the CLIPLoss method. This produces embeddings that can be directly compared for alignment between text and images. The cosine similarity between the text and image embeddings was computed for every pair in the batch, creating a similarity matrix. Cross-entropy loss was then applied to this matrix, assigning a larger loss value when an image had greater similarity with an incorrect text description than with its correct one. In this way, CLIP loss encouraged the model to align the semantic meaning between generated images and their corresponding input text.

By training the model with both Perceptual loss and CLIP loss, image quality improved significantly compared to pixel reconstruction losses. The generated outputs show recognisable geometric shapes and clearer contrast between the left

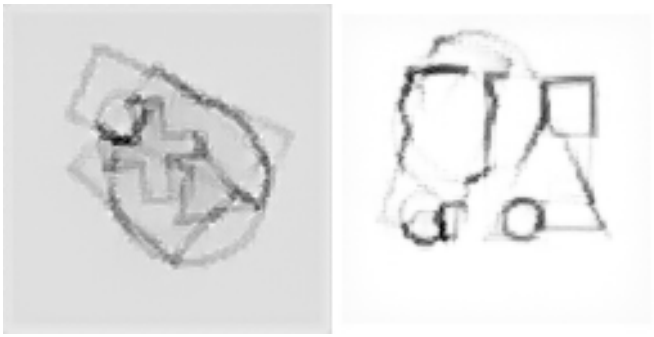

Figure 3.2: Model outputs trained with CLIP and perceptual loss for the input `"RIGHT(EXACTLY(2,FIGURES))"` to describe BP#23. While the geometric arrangements are clear, the images show superimposed shapes.

and right image pairs. However, the model generated overlapping shapes in outputs, appearing to superimpose multiple training examples rather than extracting the underlying visual concept as shown in figure 3.2. Even though CLIP and Perceptual loss improved image quality, empirical testing was conducted to evaluate the actual contribution of both loss functions. This analysis investigated whether ViT-B/32 and VGG19 were providing appropriate feature representations for the geometric arrangements found in Bongard Problems.

3.4.1 ViT-B/32 and VGG19 Analysis

VGG19 was tested first to see if it was able to distinguish between the images in Bongard Problems. The same feature layer blocks from training were used from VGG19 to extract the features from an image. Each possible unique image pair was compared for a given Bongard Problem. The cosine similarity was calculated between the extracted image features. This created a matrix of 12x12 image cosine similarities. The cosine similarities were plotted into a heatmap using matplotlib and seaborn Python libraries. Ideally the top left quadrants (left group of images) and the bottom right quadrant (right group of images) would have high cosine similarity due to sharing some underlying concept. While

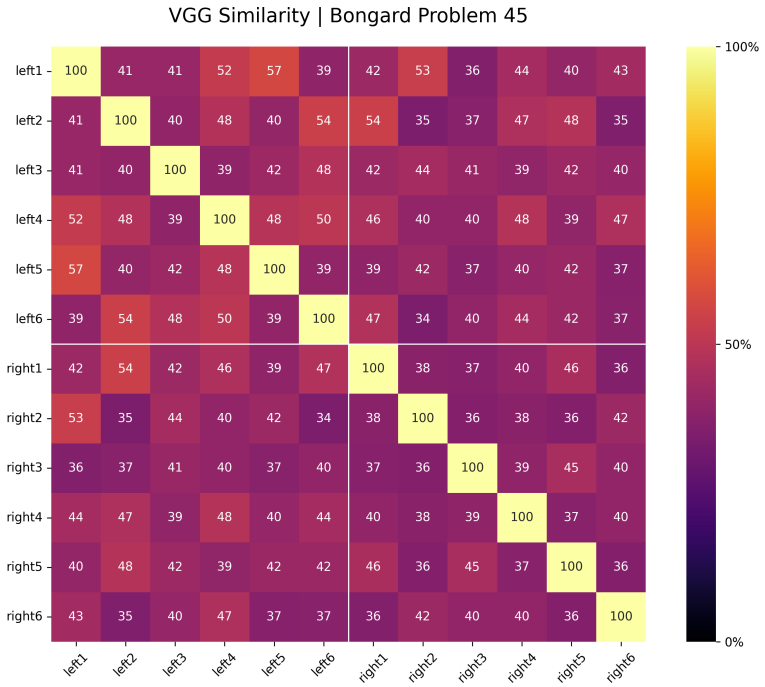

Figure 3.3: VGG19 cosine similarity for BP#45.

the other quadrants would have a low cosine similarity due to the images being in opposing groups. However, as shown in Figure 3.3, VGG19 was unable to reliably discern the image groups from each other but in rare cases it could such as BP#8 shown in figure 3.4. Analysis revealed that VGG19 provided only weak discrimination between opposing image pairs (cosine similarities ranging 0.3–0.6). However, during the training process VGG19 is used to compare the similarity between the generated image and its corresponding real image. Nevertheless, this analysis revealed how VGG19 represents geometric arrangements, offering insight into its feature capabilities.

Next, the same image pair analysis was done but by using the CLIP ViT-B/32 model. Surprisingly, ViT-B/32 produced nearly identical cosine similarities for all possible image pairs, ranging from 0.9-1.0 as shown in figure 3.5. This means that ViT-B/32 considers almost every single Bongard Problem image identical. Upon

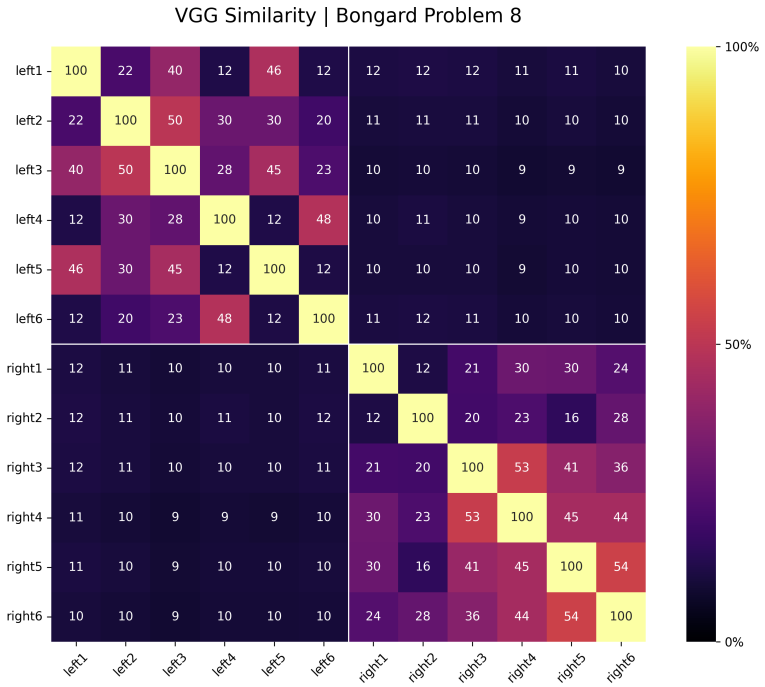

Figure 3.4: Plotted VGG19 cosine similarity for BP#8.

further analysis, this limitation stems from how CLIP models were trained. Models like ViT-B/32 were trained on 400 million real life images, meaning there was little or no distribution of geometric shapes. Therefore ViT-B/32 never learned representations which could help with discerning or recognising geometric arrangements. As a result, during training, ViT-B/32 failed to provide meaningful guidance for semantic understanding, regardless of the input text. With this information, two additional tests involving ViT-B/32 were conducted. Analysis of text-to-image pairs and text-to-text pairs using ViT-B/32 revealed almost identical cosine similarities across all comparisons. These findings confirm that ViT-B/32's feature representations are unsuitable for the Bongard Problem domain. Consequently, CLIP loss was removed from the training process, which also improved training times by reducing computational overhead due to one fewer model being involved in training. Complete sets of heatmaps for both VGG19

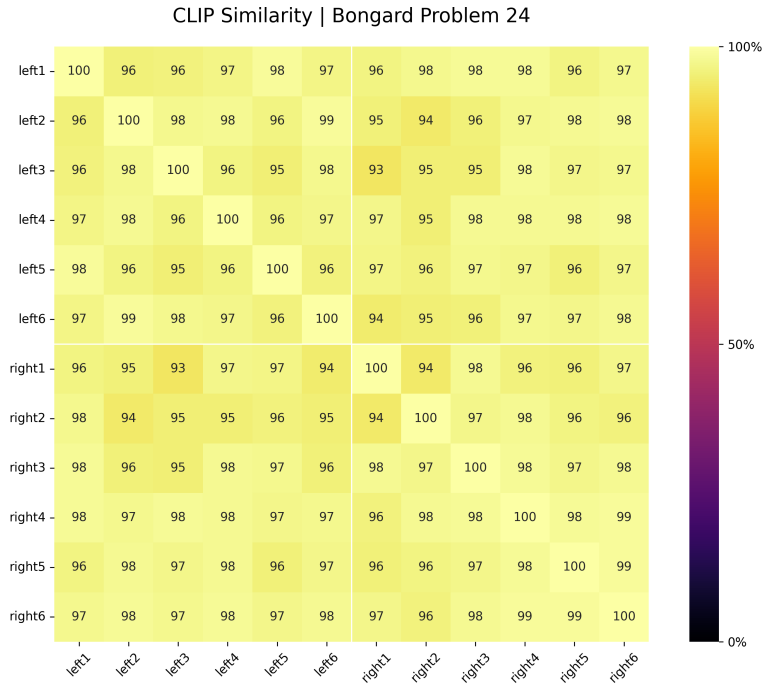

Figure 3.5: ViT-B/32 cosine similarities for BP#24.

and ViT-B/32 are available in the code repository: https://github.com/Pasha-Akito/BP_Image_generator

3.4.2 Perceptual Loss Improvements

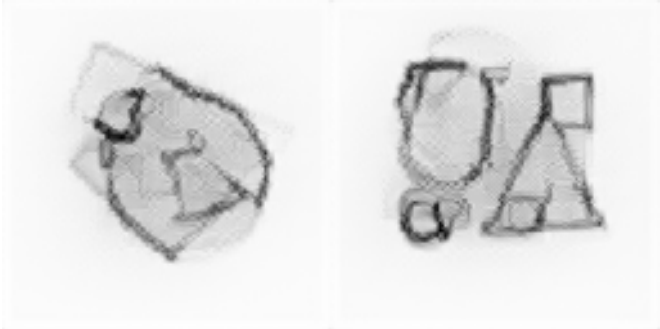

Figure 3.6: Model outputs trained with weighted perceptual loss for the input `"RIGHT(EXACTLY(2,FIGURES))"` which describes BP#23. While clearer geometric shapes are present, outputs still show superimposed training data.

Following the removal of CLIP loss, efforts were concentrated on optimising Perceptual loss parameters to improve the quality of generated geometric arrangements. The first step was to decide which VGG19 feature layers to use. Early layers capture edges and shapes, while later layers capture higher level textures more suited to identifying objects, animals and scenes [11]. Since Bongard Problems depend mainly on edge and shape information, experiments were conducted to identify which feature blocks contributed most effectively.
The following blocks were finalised after testing:

- Layers 0-2: Conv2d → ReLU → **Conv2d (Layer 2)**
- Layers 3-7: ReLU → MaxPool2d → Conv2d → ReLU → **Conv2d (Layer 7)**
- Layers 8-12: ReLU → MaxPool2d → Conv2d → ReLU → **Conv2d (Layer 12)**
- Layers 13-14: ReLU → **Conv2d (Layer 14)**
- Layers 15-16: ReLU → **Conv2d (Layer 16)**
- Layers 17-21: ReLU → MaxPool2d → Conv2d → ReLU → **Conv2d (Layer 21)**

A new hyper-parameter, FEATURE_WEIGHTS, was introduced as a multiplier on the L2 distances from each feature block. This gave greater weight to the early layers, while allowing later layers to have only a smaller influence on the overall perceptual loss. This allows the model to benefit from high level textural information from later layers while ensuring early edge and shape features dominate the perceptual loss.
While the generated outputs demonstrated clearer edges and improved object shapes, the continued superimposition of training images indicated that the model

was relying on memorisation rather than learning generalised concepts. This behaviour prompted investigation into methods that could encourage the extraction of underlying visual patterns instead of reproducing specific examples.

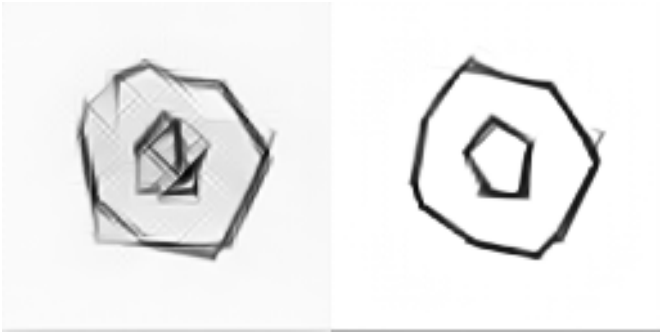

Figure 3.7: Comparison of model outputs: later VGG layers (left) vs weighted early layers (right). Note: Dropout regularisation from Section 3.5 was also applied. This comparison isolates the effect of later vs early layers.

3.5 Dropout Breakthrough

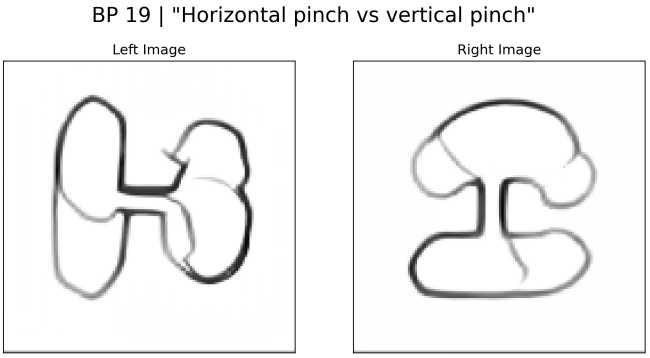

Figure 3.8: Model outputs with Dropout2d added after each ReLU layer, using BP#19 as an example. Clear geometric arrangements are visible, with no superimposed shapes.

Dropout2d was introduced after each ReLU layer in the image generator architecture. Dropout2d works by randomly deactivating channels during training. By adding in a regularisation mechanism, the model stopped relying on memorisation to minimise perceptual loss. As a result, the model started demonstrating concept learning by generating representations that embody the concepts. For example, outputs for 'horizontal pinch vs vertical pinch' and 'elongated hull vs compact hull' captured the core visual features of these geometric arrangements. As shown in figure 3.8, the generated arrangements are clear with no superimposed shapes present.

3.6 Training Instability and Fixes

Training instability was an issue during development of the model. There were two key issues:

1. One or both of the image generators would start generating grey backgrounds, or big groups of grey pixels for all outputs.

2. Early on during training, the model would converge where only noise is generated for all possible outputs for both image generators.

An initial workaround was to just start training from scratch, but this was not ideal, and not manageable when the instability grew due to adding high resolution Bongard Problem images. Early on, 104x104 Bongard Problem images were used from Yun Xinyu's github [28]. Later on, Depeweg et al's higher resolution images at 432x432 were used [18, 19]. This caused a need to fix the instability issues. Upon further investigation of the current architecture, the input images to VGG19 were not normalised correctly. Due to the Tanh output from the image generator, images were in range [-1,1] while VGG19 accepted [0,1]. When the images were normalised to [0,1] before being input into VGG19, the grey backgrounds and big groups of grey pixels disappeared from the outputs.

Next was addressing the early converging to noise issue. Every 100 batches, generated images were saved to a folder called training_debug, which gave immediate feedback on the images the model was generating during training. At around 8 epochs, the model frequently converged where only noise was generated and failed to improve image quality despite additional training epochs, even when loss continued to lower. Extensive testing was done and the issue was traced to the optimiser. The Adam optimiser with a learning rate of 0.00005 was finalised, which let the model converge on good outputs while not getting stuck generating noise.

3.7 Final Training Configuration

Following the resolution of training stability issues, the final training configuration was established to ensure reliable and reproducible model training. Training would automatically terminate at 100 epochs, though it was often stopped earlier around 70 epochs when outputs showed good representations and average loss plateaued. However, for consistency in the results presented in this thesis, all generated images were produced using models trained to the full 100 epochs. The following lists the finalised hyper-parameters used for the model whose outputs are reported in the Results chapter

- **Optimiser**: Adam with learning rate 0.00005
- **Batch Size**: 32
- **Training Epochs**: 100
- **Gradient Clipping**: 1.0
- **VGG19 feature layer blocks**: [2, 7, 12, 14, 16, 21]
- **VGG19 feature layer weights**: [1.9, 1.0, 0.5, 0.35, 0.25, 0.15]

Batch size was fixed at 32 due to hardware constraints. Gradient clipping was set to 1.0, as higher values were tested but proved less stable. Many different weights and layers were tried with VGG19, with these settings generally showing a good representation for all concepts. When middle or later layer weights are higher, it is possible to see better representations for concepts which could be considered more abstract, but generally this only improves the representations for very few Bongard Problems while all other representations suffer in quality due to missing edges or vital geometric information.

Chapter 4

Experiments

Three training experiments were conducted using the final training configuration from Chapter 3.7. Each dataset, Symbolic, English and Minimal were trained separately for 100 epochs with identical hyper-parameters. Once training was finished, test images were generated based on the Bongard Problem descriptions in the given dataset. Training was conducted on an NVIDIA GeForce RTX 4060 with 8GB VRAM, 32GB RAM and using Python 3.9.13, PyTorch 2.7.0, torchvision 0.22.0 and CUDA 12.8. The steps to reproduce are as follows, modify in config.py to point to the specific dataset, run train.py and then run generate_images.py once training is complete. Complete source code is available in the code repository: https://github.com/Pasha-Akito/BP_Image_generator

4.1 Creating Results

Subjective visual assessment by the author was conducted on all the generated image outputs across all three datasets. Three detailed result tables were created, one for each dataset which assesses each Bongard Problem available in the dataset. The detailed result tables can be found in appendix A.

Each problem is split into a left and right image assessment, as for some concepts one side of the image was very clear while the other failed to generate correctly. Three classifications exist for the image assessment, CLEAR, FAILED and PARTIAL. Where if an output is undoubtedly showcasing the concept, then it is CLEAR. PARTIAL is used rarely where part of the concept shows in the output. FAILED is when the image output does not represent the concept, or just generates noise. Finally, although the Symbolic Dataset has multiple representations for one Bongard Problem, visually identical outputs were consolidated into a single entry in the table, resulting in 75 Bongard Problem evaluations instead of 100 like for the English and Minimal Dataset.

4.1.1 Concept Type Categorisation

To answer RQ2, all 100 Bongard Problems were categorised into five main concept types:

Spatial Relationships: Concepts where specific location, orientation or positional arrangement of geometric elements is necessary to solve the problem. These concepts require understanding relative position such as "above vs below", "inside vs outside" and collinear arrangements.

Shape and Geometry: Concepts which focus on the geometric properties or structural features. These concepts require understanding shape properties such as "convex vs concave", "triangle vs quadrilateral" and "curved vs straight lines".

Numerosity: Concepts which require counting, assessing quantity or numerical comparison to solve the problem. Problems such as

"one vs two figures", "more vs fewer elements" and "three parts vs five parts" are considered numerosity.

Visual Properties: Concepts which require assessing the surface appearance, or texture of shapes rather than their geometric structure to solve. Problems involve distinguishing between "filled vs outlined", "identical vs different figures", or "same colour vs different colours" to solve.

Size: Concepts involving relative scale or dimension comparison. These problems require analysing the dimensional characteristics of shapes such as "Big vs small" or "same size vs different size" to solve.

The concept types remained identical for all three table results. The only changing variables between each table were the subjective visual assessment of image outputs, and the input text.

Chapter 5

Results

All the following visualisations and analysis of the results are based on the detailed evaluation tables which are available in appendix A. Although left and right images were assessed separately, only pairs with the same evaluation outcome are shown here for clarity.

5.1 Research Question 1

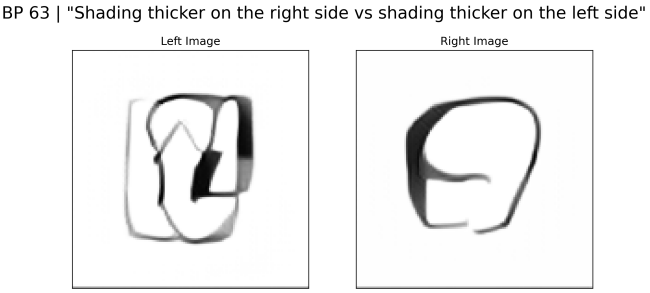

Figure 5.1: Example of CLEAR result for both images. **Input:** "Shading thicker on the right side vs shading thicker on the left side." **Dataset:** English. **BP:** #63. **Concept:** "Shading thicker on the right side vs shading thicker on the left side." **Category:** Size.

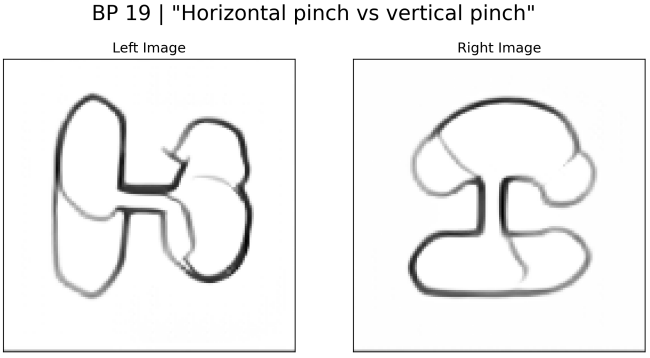

Figure 5.2: Example of CLEAR result for both images. **Input:** "Horizontal pinch vs vertical pinch." **Dataset:** English. **BP:** #19. **Concept:** "Horizontal pinch vs vertical pinch." **Category:** Shape and Geometry.

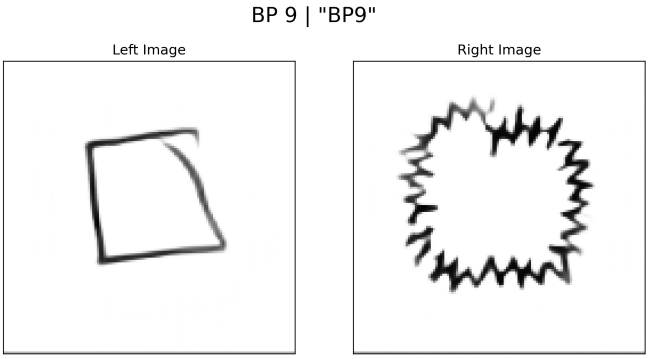

Figure 5.3: Example of CLEAR result for both images. **Input:** "BP9." **Dataset:** Minimal. **BP:** #9. **Concept:** "Normal outline vs wiggly outline." **Category:** Shape and Geometry.

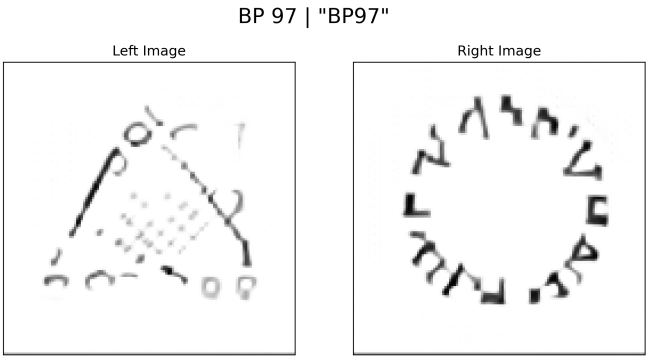

Figure 5.4: Example of CLEAR result for both images. **Input:** "BP97." **Dataset:** Minimal. **BP:** #97. **Concept:** "Triangles vs Circles." **Category:** Shape and Geometry.

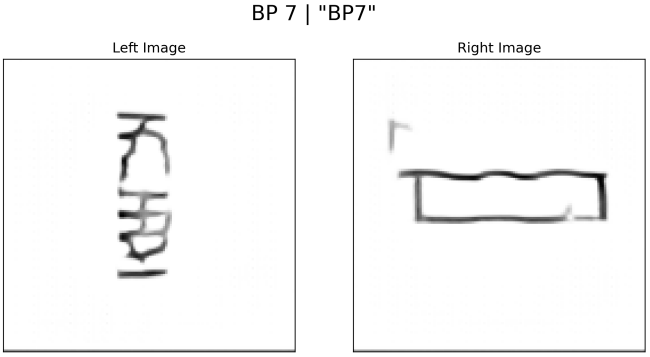

Figure 5.5: Example of CLEAR result for both images. **Input:** "BP7." **Dataset:** Minimal. **BP:** #7. **Concept:** "Taller than wide vs. wider than tall." **Category:** Size.

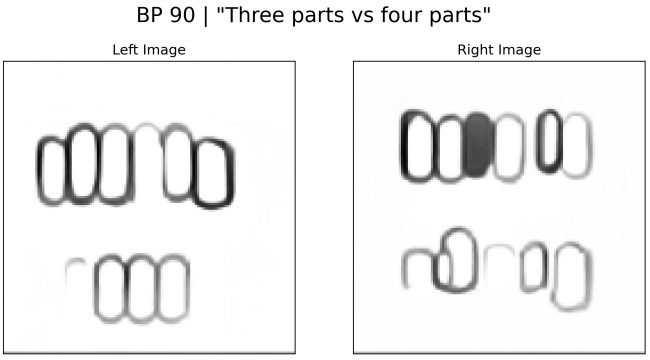

Figure 5.6: Example of PARTIAL result for both images. **Input:** "Three parts vs four parts." **Dataset:** English. **BP:** #90. **Concept:** "Three parts vs four parts." **Category:** Numerosity. Both images receive a PARTIAL because there is almost three groups vs four groups, but the noise present inbetween these groups makes it hard to mark them distinct enough to be CLEAR status.

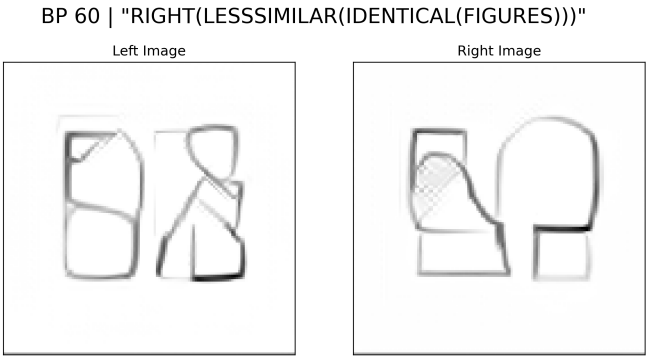

Figure 5.7: Example of PARTIAL result for both images. **Input:** `RIGHT(LESSSIMILAR(IDENTICAL(FIGURES)))`. **Dataset:** Symbolic. **BP:** #60. **Concept:** "Some similar figures vs. no similar figures." **Category:** Visual Properties. The left hand side receives a PARTIAL because while the two distinct shapes are somewhat similar, they are not similar enough (in the authors opinion) to warrant a CLEAR result. The right image receives PARTIAL because while there is a big vs small shape, the noise on the right shape makes it hard to confidently say the model learned the concept.

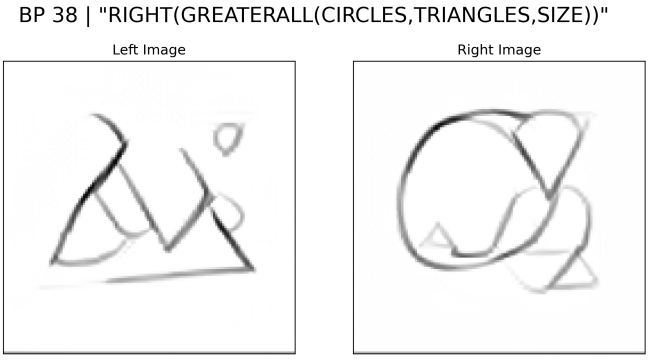

Figure 5.8: Example of PARTIAL result for both images. **Input:** `RIGHT(GREATERALL(CIRCLES,TRIANGLES,SIZE))`. **Dataset:** Symbolic. **BP:** #38. **Concept:** "Triangle larger than circle vs. triangle smaller than circle." **Category:** Size. The left image received a PARTIAL because a small circle does exist, but fails to generate a correct Triangle for comparison. The same is said for the right image, a large circle is generated, but a hard to identify triangle makes it PARTIAL.

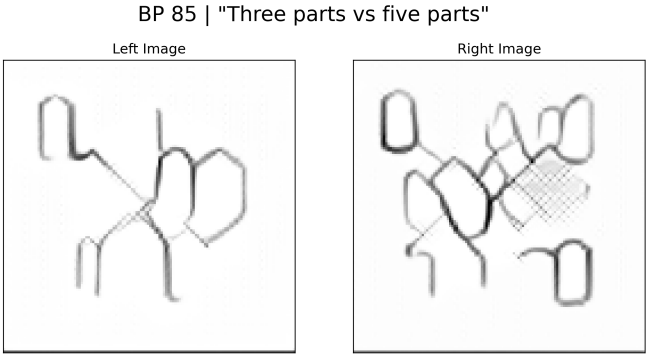

Figure 5.9: Example of FAILED result for both images. **Input:** "Three parts vs five parts." **Dataset:** English. **BP:** #85. **Concept:** "Three parts vs five parts." **Category:** Numerosity.

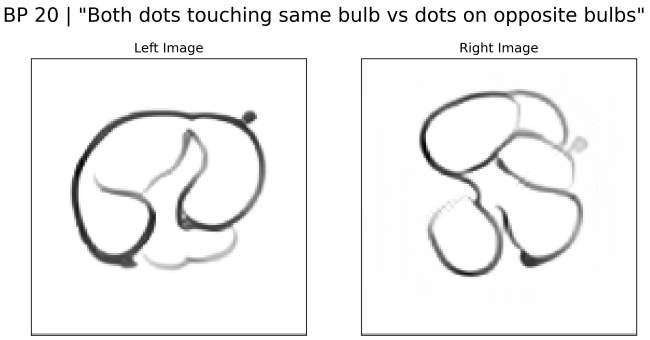

Figure 5.10: Example of FAILED result for both images. **Input:** "Both dots touching same bulb vs dots on opposite bulbs." **Dataset:** English. **BP:** #20. **Concept:** "Both dots touching same bulb vs dots on opposite bulbs." **Category:** Spatial Relationship.

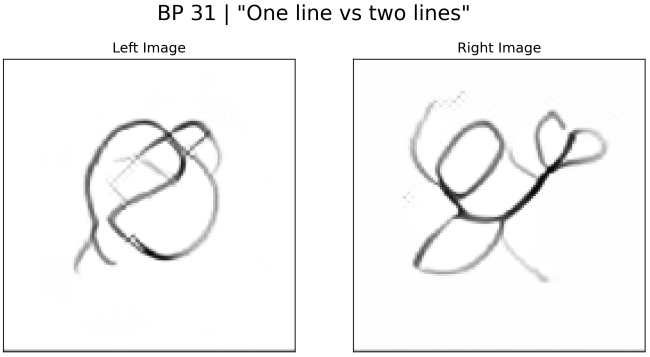

Figure 5.11: Example of FAILED result for both images. **Input:** "One line vs two lines." **Dataset:** English. **BP:** #31. **Concept:** "One line vs two lines." **Category:** Numerosity.

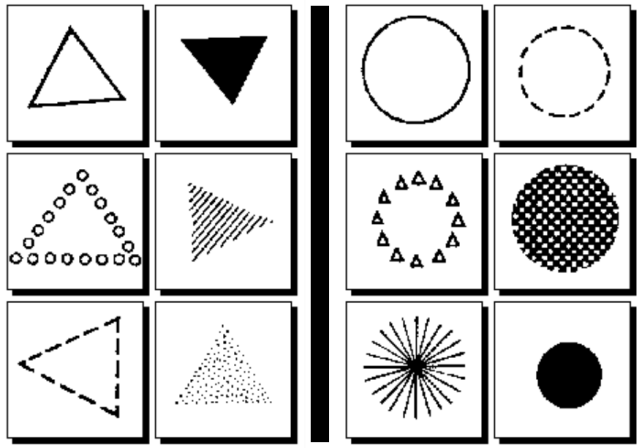

Figure 5.12: Showcasing full BP#97 to compare model representation. Images from foundalis.com [1] by Mikhail M. Bongard [2]

RQ1 asks how neural networks learn visual concepts when trained exclusively with perceptual loss? The presented outputs demonstrate that the model is capable of learning the underlying concept through perceptual loss alone. However, the model only does so because Dropout2d provided regularisation which prevented memorisation of the training data. Without Dropout2d, the model would start memorising all the data, leading to superimposed shapes from training data in outputs. With Dropout2d, the model has learned representations which are consistent between all training images, therefore demonstrating the ability to learn visual concepts.

For example BP#97 as shown in figure 5.12 consists of images with very different textures or compositions to create large triangles on the left, and large circles on the right. The model was able to create a representation for BP#97 as shown in figure 5.4 where a single large triangle exists on the left, and a single large circle exists on the right, irrespective of the different textures or compositions present in the original problem. This section only shows a handful of representations generated by the model, all representations for all datasets can be found in the code repository https://github.com/Pasha-Akito/BP_Image_generator

5.2 Research Question 2

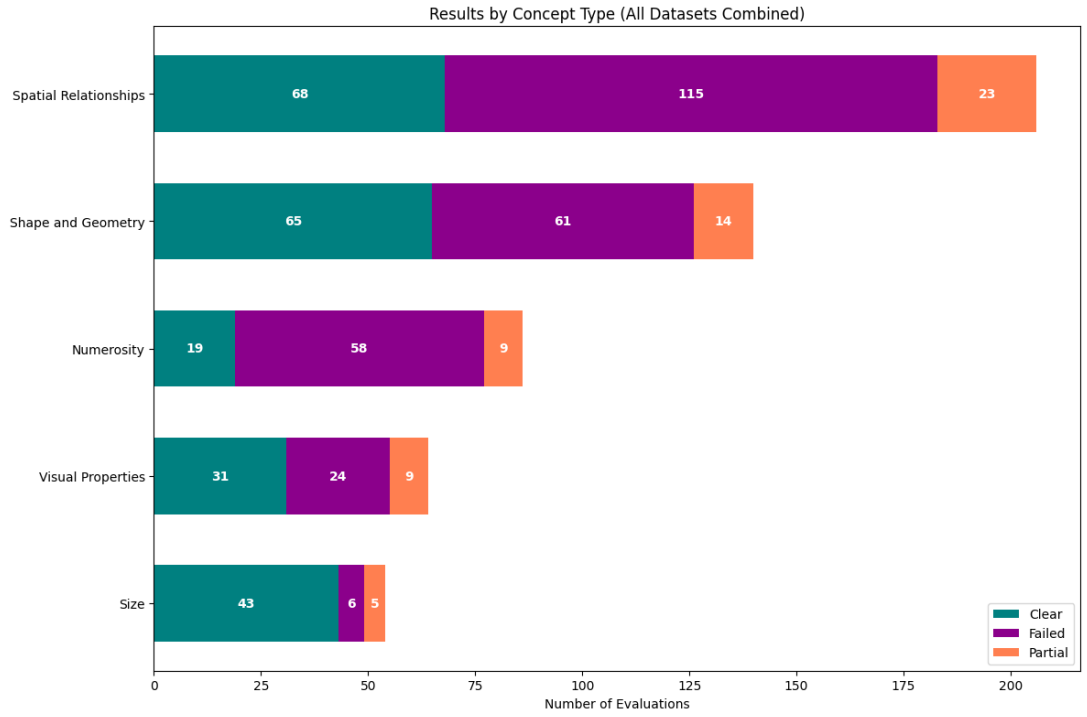

Figure 5.13: Showcasing output results per concept type for all datasets combined. Plot is created from data from tables found in appendix.

Concept Learning Success Rates:

- **Size**: 43/54 clear results (80% success rate)
- **Visual Properties**: 31/64 clear results (48% success rate)
- **Shape and Geometry**: 65/140 clear results (46% success rate)
- **Spatial Relationships**: 68/206 clear results (33% success rate)
- **Numerosity**: 19/86 clear results (22% success rate)

RQ2 asks which types of visual concepts are more naturally learnable through perceptual similarity alone? Analysis of concept learning success rates across all datasets reveal patterns in ease of learnability:

Concepts which involve Size have the highest success rate at 80%. This indicates size is the most naturally learnable through perceptual similarity alone. Visual properties and shape/geometry concepts show moderate performance at 48% and 46% respectively, suggesting these visual features can be partially captured through perceptual loss but with mixed results. Conversely, concepts involving numerosity show the lowest success rate at 22%, suggesting that counting tasks require more than just perceptual similarity to learn well. Spatial relationships had the most clear results with 68 but demonstrate a low success rate at 33%. This indicates that there is high variability of difficulty in this category. Some spatial concepts align well with perceptual similarity while others would require more complex relational understanding. The findings show that perceptual loss performs well at learning concepts based on measurable visual features, but struggles with more nuanced tasks such as counting and abstract concepts which require relational understanding without clear visual comparison.

5.3 Research Question 3

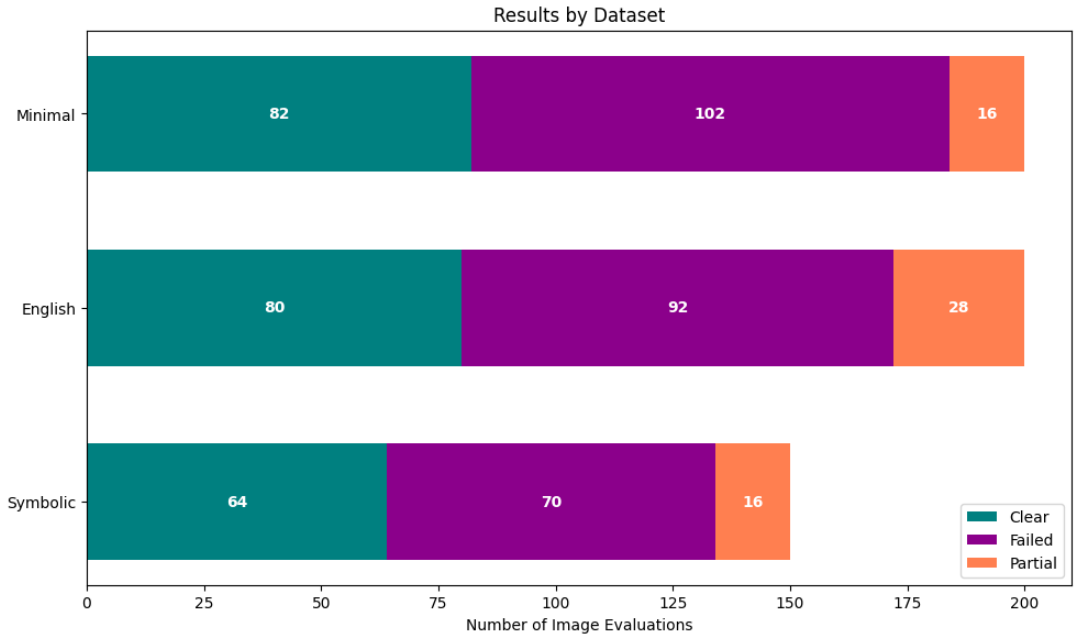

Figure 5.14: Showcasing output results per dataset. Plot is created from data from tables found in appendix.

Success Rates by Dataset:

- **Symbolic**: 64/150 clear results (43% success rate)
- **Minimal**: 82/200 clear results (41% success rate)
- **English**: 80/200 clear results (40% success rate)

Failure Rates by Dataset:

- **English**: 28 partial results (14% partial rate), 92 failed results (46% failure rate)
- **Symbolic**: 16 partial results (10.7% partial rate), 70 failed results (46.7% failure rate)

- **Minimal**: 16 partial results (8% partial rate), 102 failed results (51% failure rate)

RQ3 asks, How does linguistic representation affect visual concept learning in generative models? Analysis of the dataset results reveal a modest impact on concept learning:

The Symbolic Dataset achieves the highest success rate (43%), followed closely by Minimal (41%) and English (40%). However, the Symbolic Dataset omitted many of the complex Bongard Problems which both Minimal and English failed to represent successfully. More revealing are the differences in the failure patterns. The English descriptions show the highest rate of partial successes at 14%, as well as the lowest failure rate 46%. Symbolic descriptions follow behind with less partial results at 10.7% and slightly higher failure rates at 46.7%. The Minimal Dataset had the highest amount of clear results at 82, however, its low partial results at 8% puts the failure rate as the highest at 51%.

This suggests that natural language, and even symbolic language, provides the model with enough semantic understanding to achieve partial concept understanding, even when it misses the entire concept. In contrast, the Minimal Dataset shows very binary outcomes, due to purely relying on visual elements. Ultimately, Perceptual similarity remains the most dominant factor in determining the success of the outputs, however, linguistic representations can affect visual concept learning modestly.

Chapter 6

Conclusion

This thesis set out to investigate how a simple transformer architecture could learn visual concepts through image generation using Bongard Problems as the testing data. The research demonstrates that simple architectures can learn visual concepts when trained exclusively with perceptual loss, provided the architecture allows regularisation which prevents memorisation of training data.

Analysis across 100 Bongard Problems revealed that the model was able to learn accurate representations of concepts, with certain concept types showing a higher success rate. Concepts involving Size achieved a 80% success rate, where as numerosity concepts only succeeded 22% of the time. The pattern indicates that measurable visual features align more naturally with perceptual loss training, whereas abstract reasoning tasks requiring counting or precise spatial relationships that demand understanding the state of geometric arrangements remain challenging.

The success rate of certain concepts align with recent evaluations of state-of-the-art models such as o1, GPT-4o, Claude 3.5 and Gemini 2.0 on the same 100 Bongard Problems [4]. Concepts involving size showed a high success rate, with o1 solving 67% of problems, whereas numerosity proved to be challenging

for all state-of-the-art models. This suggests that some visual concepts are inherently harder to learn than others, regardless of the specific architecture, training method, or scale of data used.

Additionally, analysis done across three different linguistic representations (symbolic, natural language and minimal numerical labels) showed modest differences in concept learning success (43%, 40% and 41% respectively). But more notable, symbolic and natural language had less total failures as compared to the minimal numerical identifiers (46% for English and 46.7% for Symbolic vs 51% for Minimal). Suggesting that the model did learn some semantic understanding between language and visual features enough to give more partially correct representations.

These findings contribute to understanding the fundamental nature of visual concept learning by demonstrating that concept difficulty patterns emerge consistently across different architectures and training approaches. This thesis shows that even simple transformer architectures trained solely with perceptual loss can extract meaningful visual concepts from limited data when appropriate regularisation methods prevent memorisation. Future work could focus on building a CLIP-style contrastive model trained specifically on Bongard Problems. Beyond this, this generative analysis approach could be applied to other visual reasoning datasets to test how broadly these findings hold.

References

[1] H. Foundalis, "Index of bongard problems," 1999, website with Bongard Problems and their Solution. [Online]. Available: https://www.foundalis.com/res/bps/bpidx.htm v, ix, 2, 11, 13, 39

[2] M. M. Bongard, *Pattern Recognition*. Spartan Books, 1970. v, ix, 1, 2, 11, 13, 39

[3] H. E. Foundalis, "Phaeaco: A cognitive architecture inspired by bongard's problems," Ph.D. dissertation, Indiana University, Bloomington, IN, 2006. [Online]. Available: https://www.foundalis.com/res/diss_research.html 1, 11, 13

[4] A. Wüst, T. Woydt, L. Helff, I. Ibs, W. Stammer, D. S. Dhami, C. A. Rothkopf, and K. Kersting, "Bongard in wonderland: Visual puzzles that still make ai go mad?" 2025. [Online]. Available: https://arxiv.org/abs/2410.19546 1, 44

[5] E. Rosch and C. B. Mervis, "Family resemblances: Studies in the internal structure of categories," *Cognitive psychology*, vol. 7, no. 4, pp. 573–605, 1975. 1

[6] C. Zhang, B. Jia, Z. Song-Chun, and Y. Zhu, "Abstract spatial-temporal reasoning via probabilistic abduction and execution," *arXiv.org*, 2021. 2, 3

[7] J. C. Raven, "Mental tests used in genetic studies: The performance of related individuals on tests mainly educative and mainly reproductive," MSc Thesis, University of London, 1936. 2

[8] J. C. Raven and J. H. Court, *Raven's progressive matrices and vocabulary scales*. Oxford Psychologists Press Oxford, 1998. 2

[9] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016. 8

[10] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," 2018. [Online]. Available: https://arxiv.org/abs/1611.07004 8

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, Conference paper, cited by: 29712. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85083953063&partnerID=40&md5=a5b2d6b3fc9f0a6f92864467079a1280 8, 26

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. 9

[13] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision – ECCV 2016*, ser. Lecture Notes in Computer Science, vol. 9906. Springer, 2016, pp. 694–711. 9

[14] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The

unreasonable effectiveness of deep features as a perceptual metric," 2018. [Online]. Available: https://arxiv.org/abs/1801.03924 9

[15] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021. [Online]. Available: https://arxiv.org/abs/2103.00020 9

[16] G. Kim, T. Kwon, and J. C. Ye, "Diffusionclip: Text-guided diffusion models for robust image manipulation," 2022. [Online]. Available: https://arxiv.org/abs/2110.02711 10

[17] S. Kharagorgiev, "Solving bongard problems with deep learning," 2018, journal on using transfer learning in Solving Bongard Problems. [Online]. Available: https://k10v.github.io/2018/02/25/Solving-Bongard-problems-with-deep-learning/ 11

[18] S. Depeweg, C. A. Rothkopf, and F. Jäkel, "Solving bongard problems with a visual language and pragmatic reasoning," *arXiv.org*, 2018. 11, 13, 16, 29

[19] ——, "Solving bongard problems with a visual language and pragmatic constraints," *Cognitive science: a multidisciplinary journal of artificial intelligence, psychology, and language*, vol. 48, no. 5, 2024. 11, 13, 16, 29

[20] X. Yun, T. Bohn, and C. Ling, in *Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science. Switzerland: Springer International Publishing AG, 2020, vol. 12109, pp. 528–539. 11

[21] W. Nie, Z. Yu, L. Mao, A. B. Patel, Y. Zhu, and A. Anandkumar, "Bongard-logo: A new benchmark for human-level concept learning and reasoning," *arXiv.org*, 2021. 11

[22] E. Weitnauer, R. L. Goldstone, and H. Ritter, "Perception and simulation during concept learning." *The psychological review.*, vol. 130, no. 5, pp. 1203–1238, 2023. 11

[23] F. Chollet, "On the measure of intelligence," *arXiv.org*, 2019. 11

[24] S. Youssef, M. Zečević, D. S. Dhami, and K. Kersting, "Towards a solution to bongard problems: A causal approach," *arXiv.org*, 2022. 12

[25] A. Linhares, "A glimpse at the metaphysics of bongard problems," *Artificial intelligence*, vol. 121, no. 1, pp. 251–270, 2000. 12

[26] L. C. Aaron David Fairbanks, "The on-line encyclopedia of bongard problems (oebp)," https://oebp.org. 18

[27] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," *Proceedings of Machine Learning Research*, vol. 139, p. 8748 – 8763, 2021, cited by: 16775. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85147256635&partnerID=40&md5=dcac67f6787355a316158a6bac027cd3 21

[28] X. Yun, "Bongard problems," 2024. [Online]. Available: https://github.com/XinyuYun/bongard-problems 29

Appendix A

Detailed Table Results

Table A.1: English Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
1	Empty image vs non-empty image	visual_properties	CLEAR	CLEAR
2	Big vs small	size	CLEAR	CLEAR
3	Hollow outline vs filled in solid	visual_properties	CLEAR	FAILED
4	Convex vs concave	shape_geometry	FAILED	FAILED
5	Is polygon vs is smooth without straight lines or corners	shape_geometry	CLEAR	PARTIAL
6	Triangle vs quadrilateral	shape_geometry	PARTIAL	CLEAR
7	Taller than wide vs wider than tall	size	PARTIAL	CLEAR
8	Positioned right vs positioned left	spatial_relationship	CLEAR	CLEAR
9	Non-wiggly outline vs wiggly outline	visual_properties	CLEAR	CLEAR

Continued on next page

Table A.1: English Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
10	Approximately triangular outline vs approximately convex quadrilateral outline	shape_geometry	CLEAR	CLEAR
11	Thin and elongated vs compact	shape_geometry	FAILED	CLEAR
12	Thin elongated convex hull vs compact convex hull	shape_geometry	CLEAR	CLEAR
13	Tall rectangle OR wide ellipse vs wide rectangle OR tall ellipse	shape_geometry	FAILED	FAILED
14	All big individual figures vs all small individual figures	size	CLEAR	CLEAR
15	Closed shape outline vs non-closed curve	shape_geometry	CLEAR	FAILED
16	Clockwise spiraling curve vs counter-clockwise spiraling curve	shape_geometry	FAILED	CLEAR
17	Shape with a reflex corner vs shape without a reflex corner	shape_geometry	CLEAR	CLEAR
18	'Pinched' shape (drastically thinner somewhere in the middle than on the ends) vs non-pinched shape	shape_geometry	PARTIAL	PARTIAL
19	Horizontal pinch vs vertical pinch	shape_geometry	CLEAR	CLEAR
20	Both dots touching same bulb vs dots on opposite bulbs	spatial_relationship	FAILED	FAILED

Continued on next page

Table A.1: English Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
21	Small shape present vs all shapes large	size	CLEAR	CLEAR
22	All shapes approximately the same size vs shapes of different size	size	PARTIAL	CLEAR
23	One vs two figures	numerosity	FAILED	FAILED
24	A circle vs no circle	numerosity	CLEAR	CLEAR
25	Black figure is a triangle vs black figure is a circle	visual_properties	PARTIAL	FAILED
26	Solid black triangle vs no solid black triangle	numerosity	FAILED	PARTIAL
27	More solid black figures vs more outline figures	numerosity	CLEAR	PARTIAL
28	More solid black circles vs more outline circles	numerosity	FAILED	CLEAR
29	There are more small circles inside the figure outline than outside vs there are fewer small circles inside the figure outline than outside	numerosity	FAILED	FAILED
30	A curve with one self-crossing vs a curve without a self-crossing	shape_geometry	FAILED	FAILED
31	One line vs two lines	numerosity	FAILED	FAILED
32	A sharp projection vs no sharp projection	shape_geometry	PARTIAL	CLEAR
33	Acute angle vs no acute angle	shape_geometry	CLEAR	CLEAR

Continued on next page

Table A.1: English Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
34	A large hole vs a small hole	size	CLEAR	CLEAR
35	The axis of the hole is parallel to the figure axis vs the axis of the hole is perpendicular to the figure axis	spatial_relationship	FAILED	FAILED
36	Triangle above circle vs circle above triangle	spatial_relationship	CLEAR	CLEAR
37	Triangle above circle vs circle above triangle	spatial_relationship	CLEAR	CLEAR
38	Triangle larger than circle vs triangle smaller than circle	size	PARTIAL	FAILED
39	Segments approximately parallel to each other vs large angles between segments	spatial_relationship	PARTIAL	PARTIAL
40	Three points on a straight line vs no three points on a straight line	spatial_relationship	PARTIAL	FAILED
41	Outline circles on one straight line vs outline circles not on one straight line	spatial_relationship	FAILED	FAILED
42	Points inside the figure outline are on a straight line vs points inside the figure outline are not on a straight line	spatial_relationship	CLEAR	CLEAR

Continued on next page

Table A.1: English Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
43	The vibration amplitude increases from left to right vs the vibration amplitude decreases from left to right	size	CLEAR	CLEAR
44	Small circles on different arcs vs small circles on one arc	spatial_relationship	FAILED	PARTIAL
45	Outline figure on top of solid black figure vs black figure on top of outline figure	spatial_relationship	FAILED	FAILED
46	Triangle on top of the circle vs circle on top of the triangle	spatial_relationship	CLEAR	CLEAR
47	Triangle inside of the circle vs circle inside of the triangle	spatial_relationship	FAILED	PARTIAL
48	Solid dark figures above the outline figures vs outline figures above the solid dark figures	spatial_relationship	CLEAR	CLEAR
49	Points inside the figure outline are grouped more densely than outside the contour vs points outside the figure contour are grouped more densely than inside the contour	spatial_relationship	PARTIAL	FAILED
50	Vertical axis of symmetry vs no axis of symmetry	shape_geometry	FAILED	FAILED

Continued on next page

Table A.1: English Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
51	Two circles close to each other vs no two circles close to each other	spatial_relationship	FAILED	FAILED
52	Arrows pointing in different directions vs arrows pointing in the same direction	spatial_relationship	FAILED	FAILED
53	Inside figure has fewer angles than outside figure vs inside figure has more angles than outside figure	numerosity	CLEAR	FAILED
54	A cross, circle, and triangle arranged counterclockwise vs a cross, circle, and triangle arranged clockwise	spatial_relationship	FAILED	FAILED
55	A circle is at the left of the cavity if you look from inside the figure vs a circle is at the right of the cavity if you look from inside the figure	spatial_relationship	FAILED	FAILED
56	All figures of the same color vs figures of different colors	visual_properties	CLEAR	CLEAR
57	Identical figures vs figures not identical	visual_properties	FAILED	FAILED
58	Solid dark quadrangles are identical vs solid dark quadrangles are different	visual_properties	FAILED	PARTIAL

Continued on next page

Table A.1: English Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
59	Figures are similar vs figures are not similar	visual_properties	PARTIAL	PARTIAL
60	Some similar figures vs no similar figures	visual_properties	FAILED	FAILED
61	A line separates the crosses in half vs a line does not separate the crosses in half	spatial_relationship	FAILED	FAILED
62	Ends of the curve are far apart vs ends of the curve are close together	spatial_relationship	FAILED	FAILED
63	Shading thicker on the right side vs shading thicker on the left side	size	CLEAR	CLEAR
64	A cross is located on the extension of the ellipse axis vs a circle is located on the extension of the ellipse axis	spatial_relationship	PARTIAL	PARTIAL
65	A set of triangles elongated horizontally vs a set of triangles elongated vertically	spatial_relationship	CLEAR	CLEAR
66	Unconnected circles on a horizontal line vs unconnected circles on a vertical line	spatial_relationship	FAILED	FAILED

Continued on next page

Table A.1: English Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
67	The right branch begins at a higher point than the left branch vs the right branch begins at a lower point than the left branch	spatial_relationship	CLEAR	PARTIAL
68	The end of the right branch is higher than that of the left branch vs the end of the right branch is lower than that of the left branch	spatial_relationship	CLEAR	CLEAR
69	Large black dot on the main branch vs large black dot on a side branch	spatial_relationship	FAILED	CLEAR
70	There are no side branches of the second order vs there are side branches of the second order	spatial_relationship	FAILED	FAILED
71	There are inside figures of the second order vs there are no inside figures of the second order	spatial_relationship	FAILED	CLEAR
72	Ends of the curve are parallel vs ends of the curve are perpendicular	shape_geometry	FAILED	FAILED

Continued on next page

Table A.1: English Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
73	The long axes of the ellipse and rectangle are perpendicular vs the long axes of the ellipse and rectangle are parallel	spatial_relationship	FAILED	FAILED
74	A tail grows from the obtuse end vs a tail grows from the acute end	shape_geometry	FAILED	CLEAR
75	Triangle located at the concave side of an arc vs triangle located at the convex side of an arc	spatial_relationship	FAILED	FAILED
76	Long sides concave vs long sides convex	shape_geometry	FAILED	PARTIAL
77	Angle divided in half vs angle not divided in half	shape_geometry	FAILED	FAILED
78	Extensions of segments cross at one point vs extensions of segments do not cross at one point	spatial_relationship	FAILED	CLEAR
79	A dark circle is closer to the outline circle than to the triangle vs a dark circle is closer to the triangle than to the outline circle	spatial_relationship	PARTIAL	FAILED

Continued on next page

Table A.1: English Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
80	Points located at the same distances from a cross vs points located at different distances from a cross	spatial_relationship	FAILED	FAILED
81	Dark figures can be divided from outline figures by a straight line vs convex hulls of filled and outlined figures overlap	spatial_relationship	FAILED	FAILED
82	The convex hull of the crosses forms an equilateral triangle vs the convex hull of the crosses does not form an equilateral triangle	spatial_relationship	FAILED	CLEAR
83	A circle is inside of a figure made by crosses vs a circle is outside of figures made by crosses	spatial_relationship	CLEAR	CLEAR
84	A quadrangle is outside of a figure made by circles vs a quadrangle is inside of a figure made by circles	spatial_relationship	CLEAR	CLEAR
85	Three parts vs five parts	numerosity	FAILED	FAILED
86	Three parts vs five parts	numerosity	FAILED	FAILED
87	Four parts vs five parts	numerosity	FAILED	FAILED
88	Three parts vs five parts	numerosity	FAILED	FAILED
89	Three parts vs five parts	numerosity	FAILED	FAILED

Continued on next page

Table A.1: English Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
90	Three parts vs four parts	numerosity	PARTIAL	PARTIAL
91	Three identical elements vs four identical elements	numerosity	FAILED	FAILED
92	The chain does not branch vs the chain branches	shape_geometry	CLEAR	CLEAR
93	Branches at outlined circle vs branches at solid dark circle	visual_properties	CLEAR	CLEAR
94	Solid dark circle not at end vs solid dark circle at end	visual_properties	CLEAR	CLEAR
95	Vertical hatched lines vs horizontal hatched lines	visual_properties	CLEAR	CLEAR
96	Triangles vs quadrangles	shape_geometry	FAILED	FAILED
97	Triangles vs circles	shape_geometry	CLEAR	CLEAR
98	Triangles vs quadrangles	shape_geometry	FAILED	FAILED
99	Outlines made by triangles and circles intersect vs outlines made by triangles and circles do not intersect	spatial_relationship	FAILED	PARTIAL
100	The letter A vs the letter b	shape_geometry	CLEAR	CLEAR

Table A.2: symbolic Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
2	LEFT(EXISTS(HIGH(FIGURES,SIZE)))	size	CLEAR	CLEAR

Continued on next page

Table A.2: symbolic Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
3	LEFT(EXACTLY(1,OUTLINE(FIGURES)))	visual_properties	CLEAR	FAILED
4	RRIGHT(LESSER(FIGURES,CONVEXITY))	shape_geometry	FAILED	FAILED
5	RIGHT(EXACTLY(1,GET(FIGURES,NSIDES)))	shape_geometry	CLEAR	CLEAR
6	RIGHT(EXACTLY(1,QUADRILATERALS))	shape_geometry	FAILED	FAILED
7	LEFT(GREATER(FIGURES,ORIENTATION))	size	CLEAR	CLEAR
8	LEFT(GREATER(FIGURES,XPOS))	spatial_relationship	CLEAR	CLEAR
9	RIGHT(EXISTS(NOISY(FIGURES)))	visual_properties	CLEAR	CLEAR
10	LEFT(EXACTLY(1,NOISY(TRIANGLES)))	shape_geometry	CLEAR	PARTIAL
11	LEFT(EXISTS(HIGH(FIGURES,ELONGATION)))	shape_geometry	PARTIAL	CLEAR
12	LEFT(EXISTS(HIGH(GET(FIGURES,HULL),ASPECTR))	shape_geometry	PARTIAL	CLEAR
13	LEFT(OR(EXISTS(LOW(CIRCLES,ORIENTATION))),EXISTS(HIGH(QUADRILATERALS,ORIENTATION)))	shape_geometry	FAILED	FAILED

Continued on next page

Table A.2: symbolic Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
14	RIGHT(EXISTS(LOW(FIGURES,SIZE)))	size	CLEAR	CLEAR
15	RIGHT(EXISTS(LINES))	shape_geometry	CLEAR	FAILED
16	LEFT(EXISTS(MOVING(LINES,CLOCKWISE)))	shape_geometry	CLEAR	CLEAR
17	LEFT(EXISTS(REFLEX(ANGLE)))	shape_geometry	CLEAR	CLEAR
18	LEFT(EXISTS(PINCHED(FIGURES)))	shape_geometry	FAILED	PARTIAL
19	RIGHT(EXISTS(VERTICAL(PINCHED(FIGURES))))	shape_geometry	CLEAR	CLEAR
21	LEFT(EXISTS(LOW(FIGURES,SIZE)))	size	CLEAR	CLEAR
22	RIGHT(DIFFERENT(FIGURES,SIZE))	size	FAILED	CLEAR
23	LEFT(EXACTLY(1,FIGURES))	numerosity	CLEAR	FAILED
24	LEFT(EXISTS(CIRCLES))	numerosity	CLEAR	CLEAR
25	LEFT(EXISTS(SOLID(TRIANGLES)))	visual_properties	FAILED	FAILED
26	LEFT(EXISTS(SOLID(TRIANGLES)))	numerosity	FAILED	PARTIAL
27	LEFT(LESS(OUTLINE(FIGURES),SOLID(FIGURES)))	numerosity	FAILED	CLEAR
28	LEFT(MORE(SOLID(CIRCLES),OUTLINE(CIRCLES)))	numerosity	FAILED	CLEAR

Continued on next page

Table A.2: symbolic Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
29	LEFT(LESS(CIRCLES,INSIDE(CIRCLES)))	numerosity	FAILED	FAILED
30	LEFT(EXISTS(CROSSING(LINES)))	shape_geometry	FAILED	CLEAR
31	LEFT(OR(EXACTLY(1,LINES),EXACLTY(1,CIRCLES))	numerosity	FAILED	FAILED
32	LEFT(EXISTS(MOVING(LOW(GET(FIGURES,ANGLES))),OUTWARDS))	shape_geometry	FAILED	CLEAR
33	LEFT(EXISTS(ACUTE(ANGLES)))	shape_geometry	CLEAR	CLEAR
34	LEFT(EXISTS(BIG(GET(FIGURES,HOLES))))	size	CLEAR	FAILED
35	LEFT(MORESIMILAR(FIGURES,GET(FIGURES,HOLES),ORIENTATION))	spatial_relationship	FAILED	FAILED
36	LEFT(GREATERALL(TRIANGLES,CIRCLES,YPOS))	spatial_relationship	CLEAR	CLEAR
37	LEFT(GREATERALL(TRIANGLES,CIRCLES,YPOS))	spatial_relationship	CLEAR	CLEAR
38	RIGHT(GREATERALL(CIRCLES,TRIANGLES,SIZE))	size	PARTIAL	CLEAR
39	LEFT(MORESIMILAR(LINES,ORIENTATION))	spatial_relationship	FAILED	PARTIAL

Continued on next page

Table A.2: symbolic Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
40	LEFT(EXISTS(ALIGNED(CIRCLES)))	spatial_relationship	PARTIAL	FAILED
41	LEFT(EXISTS(ALIGNED(CIRCLES)))	spatial_relationship	FAILED	FAILED
42	LEFT(EXISTS(ALIGNED(I NSIDE(FIGURES))))	spatial_relationship	CLEAR	FAILED
43	LEFT(EXISTS(SIZE(DECR EASING(MOVING(FIGUR ES,WEST)))))	size	CLEAR	CLEAR
45	LEFT(EXISTS(ON(OUTLI NE(FIGURES),SOLID(FIG URES))))	spatial_relationship	FAILED	FAILED
46	LEFT(EXISTS(ON(TRIAN GLES,CIRCLES)))	spatial_relationship	FAILED	CLEAR
47	LEFT(EXISTS(INSIDE(TRI ANGLES)))	spatial_relationship	FAILED	CLEAR
48	LEFT(LESSERALL(OUTLI NE(FIGURES),SOLID(FIG URES),YPOS))	spatial_relationship	CLEAR	CLEAR
49	LEFT(GREATERALL(CIR CLES,INSIDE(CIRCLES),D ISTANCE))	spatial_relationship	CLEAR	FAILED
50	LEFT(EXISTS(SYMMETR Y(FIGURES,VERTICAL)))	shape_geometry	FAILED	FAILED
51	LEFT(EXISTS(LOW(CIRC LES,DISTANCE)))	spatial_relationship	FAILED	FAILED

Continued on next page

Table A.2: symbolic Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
53	LEFT(GREATERALL(FIGURES,INSIDE(FIGURES),NCORNERS))	numerosity	CLEAR	CLEAR
56	LEFT(OR(EXISTS(OUTLINE(FIGURES)),EXISTS(SOLID(FIGURES))))	visual_properties	CLEAR	CLEAR
57	LEFT(EXISTS(IDENTICAL(FIGURES)))	visual_properties	FAILED	FAILED
58	RIGHT(EXISTS(DIFFERENT(SOLID(RECTANGLES),SIZE))	visual_properties	FAILED	PARTIAL
59	RIGHT(LESSSIMILAR(IDENTICAL(FIGURES)))	visual_properties	CLEAR	PARTIAL
60	RIGHT(LESSSIMILAR(IDENTICAL(FIGURES)))	visual_properties	PARTIAL	PARTIAL
62	LEFT(EXISTS(HIGH(GET(LINES,END),DISTANCE)))	spatial_relationship	FAILED	FAILED
63	RIGHT(EXISTS(BIG(FIGURES,WEST,HULL)))	size	CLEAR	CLEAR
65	LEFT(EXISTS(ALIGNED(TRIANGLES,HORIZONTAL)))	spatial_relationship	CLEAR	PARTIAL
66	RIGHT(EXISTS(ALIGNED(CRICLES,VERTICAL)))	spatial_relationship	FAILED	FAILED
71	LEFT(EXISTS(INSIDE(INSIDE(FIGURES))))	spatial_relationship	FAILED	FAILED

Continued on next page

Table A.2: symbolic Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
72	LEFT(EXISTS(IDENTICAL(GET(LINES,END),ORIENTATION)))	shape_geometry	FAILED	FAILED
73	LEFT(EXISTS(DIFFERENT(CIRCLES,QUADRILATERALS,ORIENTATION))	spatial_relationship	FAILED	FAILED
76	RIGHT(GREATER(FIGURES,CONCAVITY))	shape_geometry	FAILED	FAILED
77	LEFT(EXISTS(IDENTICAL(GET(FIGURES,END),DISTANCE)))	shape_geometry	FAILED	FAILED
79	LEFT(AND(EXISTS(LOW(CIRCLES,SOLID(CIRCLES),DISTANCE)),EXISTS(HIGH(CIRCLES,TRIANGLES,DISTANCE))))	spatial_relationship	CLEAR	FAILED
80	LEFT(EXISTS(IDENTICAL(STARS,CIRCLES,DISTANCE)))	spatial_relationship	FAILED	FAILED
82	LEFT(EXISTS(COMPOSED(TRIANGLES)))	spatial_relationship	FAILED	CLEAR
83	LEFT(EXISTS(INSIDE(COMPOSED(FIGURES),CIRCLES))	spatial_relationship	CLEAR	CLEAR
84	RIGHT(EXISTS(INSIDE(COMPOSED(FIGURES),QUADRILATERALS))	spatial_relationship	CLEAR	FAILED

Continued on next page

Table A.2: symbolic Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
85	LEFT(OR(EXACLTY(3,GET(FIGURES,NSIDES)),EXACTLY(3,LINES)))	numerosity	FAILED	FAILED
87	LEFT(OR(EXACLTY(5,GET(FIGURES,NSIDES)),EXACTLY(5,LINES)))	numerosity	FAILED	FAILED
88	RIGHT(EXACTLY(5,CIRCLES))	numerosity	FAILED	FAILED
89	RIGHT(EXACTLY(5,COMPOSED(CIRCLES)))	numerosity	FAILED	PARTIAL
90	LEFT(EXACTLY(3,COMPOSED(OUTLINE(CIRCLES))))	numerosity	PARTIAL	FAILED
92	LEFT(EXACLTY(1,COMPOSED(LINES)))	shape_geometry	CLEAR	CLEAR
99	LEFT(ON(COMPOSED(CIRCLES),COMPOSED(TRIANGLES)))	spatial_relationship	FAILED	PARTIAL

Table A.3: minimal Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
1	BP1	visual_properties	CLEAR	CLEAR
2	BP2	size	CLEAR	CLEAR
3	BP3	visual_properties	FAILED	FAILED
4	BP4	shape_geometry	FAILED	FAILED

Continued on next page

Table A.3: minimal Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
5	BP5	shape_geometry	CLEAR	PARTIAL
6	BP6	shape_geometry	FAILED	PARTIAL
7	BP7	size	CLEAR	CLEAR
8	BP8	spatial_relationship	CLEAR	CLEAR
9	BP9	visual_properties	CLEAR	CLEAR
10	BP10	shape_geometry	CLEAR	PARTIAL
11	BP11	shape_geometry	CLEAR	CLEAR
12	BP12	shape_geometry	CLEAR	CLEAR
13	BP13	shape_geometry	FAILED	FAILED
14	BP14	size	CLEAR	CLEAR
15	BP15	shape_geometry	CLEAR	FAILED
16	BP16	shape_geometry	FAILED	CLEAR
17	BP17	shape_geometry	CLEAR	CLEAR
18	BP18	shape_geometry	FAILED	PARTIAL
19	BP19	shape_geometry	CLEAR	CLEAR
20	BP20	spatial_relationship	FAILED	PARTIAL
21	BP21	size	CLEAR	CLEAR
22	BP22	size	FAILED	CLEAR
23	BP23	numerosity	CLEAR	FAILED
24	BP24	numerosity	CLEAR	CLEAR
25	BP25	visual_properties	FAILED	FAILED
26	BP26	numerosity	FAILED	PARTIAL
27	BP27	numerosity	FAILED	CLEAR
28	BP28	numerosity	FAILED	FAILED
29	BP29	numerosity	FAILED	FAILED
30	BP30	shape_geometry	FAILED	CLEAR
31	BP31	numerosity	FAILED	FAILED

Continued on next page

Table A.3: minimal Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
32	BP32	shape_geometry	FAILED	CLEAR
33	BP33	shape_geometry	CLEAR	CLEAR
34	BP34	size	CLEAR	FAILED
35	BP35	spatial_relationship	FAILED	FAILED
36	BP36	spatial_relationship	FAILED	FAILED
37	BP37	spatial_relationship	CLEAR	CLEAR
38	BP38	size	PARTIAL	FAILED
39	BP39	spatial_relationship	PARTIAL	PARTIAL
40	BP40	spatial_relationship	PARTIAL	FAILED
41	BP41	spatial_relationship	FAILED	FAILED
42	BP42	spatial_relationship	FAILED	FAILED
43	BP43	size	CLEAR	CLEAR
44	BP44	spatial_relationship	FAILED	PARTIAL
45	BP45	spatial_relationship	FAILED	FAILED
46	BP46	spatial_relationship	FAILED	FAILED
47	BP47	spatial_relationship	FAILED	CLEAR
48	BP48	spatial_relationship	CLEAR	CLEAR
49	BP49	spatial_relationship	PARTIAL	FAILED
50	BP50	shape_geometry	FAILED	FAILED
51	BP51	spatial_relationship	CLEAR	CLEAR
52	BP52	spatial_relationship	FAILED	FAILED
53	BP53	numerosity	CLEAR	FAILED
54	BP54	spatial_relationship	FAILED	FAILED
55	BP55	spatial_relationship	CLEAR	FAILED
56	BP56	visual_properties	FAILED	CLEAR
57	BP57	visual_properties	FAILED	FAILED
58	BP58	visual_properties	FAILED	CLEAR

Continued on next page

Table A.3: minimal Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
59	BP59	visual_properties	CLEAR	PARTIAL
60	BP60	visual_properties	FAILED	FAILED
61	BP61	spatial_relationship	FAILED	FAILED
62	BP62	spatial_relationship	FAILED	FAILED
63	BP63	size	CLEAR	CLEAR
64	BP64	spatial_relationship	CLEAR	FAILED
65	BP65	spatial_relationship	CLEAR	CLEAR
66	BP66	spatial_relationship	FAILED	FAILED
67	BP67	spatial_relationship	CLEAR	CLEAR
68	BP68	spatial_relationship	CLEAR	CLEAR
69	BP69	spatial_relationship	FAILED	CLEAR
70	BP70	spatial_relationship	FAILED	FAILED
71	BP71	spatial_relationship	FAILED	FAILED
72	BP72	shape_geometry	FAILED	FAILED
73	BP73	spatial_relationship	FAILED	FAILED
74	BP74	shape_geometry	CLEAR	CLEAR
75	BP75	spatial_relationship	PARTIAL	CLEAR
76	BP76	shape_geometry	FAILED	FAILED
77	BP77	shape_geometry	FAILED	FAILED
78	BP78	spatial_relationship	FAILED	FAILED
79	BP79	spatial_relationship	FAILED	FAILED
80	BP80	spatial_relationship	CLEAR	FAILED
81	BP81	spatial_relationship	FAILED	FAILED
82	BP82	spatial_relationship	FAILED	CLEAR
83	BP83	spatial_relationship	CLEAR	CLEAR
84	BP84	spatial_relationship	CLEAR	CLEAR
85	BP85	numerosity	FAILED	FAILED

Continued on next page

Table A.3: minimal Analysis Results

BP	Text Input	Concept Type	Left Image Result	Right Image Result
86	BP86	numerosity	FAILED	FAILED
87	BP87	numerosity	FAILED	FAILED
88	BP88	numerosity	CLEAR	FAILED
89	BP89	numerosity	FAILED	CLEAR
90	BP90	numerosity	PARTIAL	FAILED
91	BP91	numerosity	FAILED	FAILED
92	BP92	shape_geometry	CLEAR	CLEAR
93	BP93	visual_properties	CLEAR	CLEAR
94	BP94	visual_properties	CLEAR	FAILED
95	BP95	visual_properties	CLEAR	CLEAR
96	BP96	shape_geometry	FAILED	FAILED
97	BP97	shape_geometry	CLEAR	CLEAR
98	BP98	shape_geometry	FAILED	FAILED
99	BP99	spatial_relationship	FAILED	PARTIAL
100	BP100	shape_geometry	CLEAR	CLEAR