DISSERTATIONES
INFORMATICAE
UNIVERSITATIS
TARTUENSIS
80

TARUN KHAJURIA

Scene understanding in human and computer vision

DISSERTATIONES INFORMATICAE UNIVERSITATIS TARTUENSIS
80

DISSERTATIONES INFORMATICAE UNIVERSITATIS TARTUENSIS
80

TARUN KHAJURIA

Scene understanding in human and computer vision

UNIVERSITY OF TARTU
Press
1632

Institute of Computer Science, Faculty of Science and Technology, University of Tartu, Estonia.

Dissertation has been accepted for the commencement of the degree of Doctor of Philosophy (PhD) in Computer Science on March 31, 2026 by the Council of the Institute of Computer Science, University of Tartu.

Supervisor

Assoc. Prof.	Jaan Aru University of Tartu Estonia

Opponents

Prof.	Tim Kietzmann University of Osnabrück Germany
Assist. Prof.	Stéphane Deny Aalto University Finland

The public defense will take place on April 28, 2026 at 10:15 in Narva Rd. 18-1017.

The publication of this dissertation was financed by the Institute of Computer Science, University of Tartu.

ISSN 2613-5906 (print) ISSN 2806-2345 (pdf)
ISBN 978-9908-57-181-2 (print) ISBN 978-9908-57-182-9 (pdf)

University of Tartu Press
http://www.tyk.ee/

To my family and friends

ABSTRACT

Humans have the ability to flexibly interpret the same visual scene in multiple ways. For example, in a cinema hall, we can identify individual seats as chairs, bean bags, or couches, while also perceiving them as part of the larger structure of rows and sections that define walkable paths. This flexibility also facilitates the robustness of our scene perception ability under challenging conditions to infer missing information, but also to disentangle relevant objects from co-occurring elements based on the context.

In this thesis, we explore the computational mechanisms that support such robustness in human vision. To understand the computational principles underlying this robustness, it is necessary to model these processes formally, allowing us to test which specific algorithmic strategies, such as iterative search or hierarchical grouping, best replicate human-like scene understanding. With this aim, we designed a challenging vision task inspired by star constellations, where outlines of objects are hidden within sparse arrangements of dots. This task serves as a controlled proxy for complex scene understanding, as it strips away low-level texture and color, forcing the visual system to rely on structural inference. In our experiments with human participants, we observed that recognizing these highly underspecified images can involve forming and iteratively refining multiple object hypotheses guided by shape and structural cues. We compare these human strategies of iterative hypothesis refinement with our proposed generative search models and popular deep learning architectures like resnet18 and pix2pix while finding objects in constellation images. This helped us distil key computational components such as multi-hypothesis search and guidance by structural fitness that can support visual inference under difficult conditions. These insights motivated our examination of structured scene representation in machine vision encoders, focusing on object binding and disentanglement in token spaces while encoding multi-object natural scenes. This examination revealed that the output of these vision encoders represent objects differently based on their importance to the scene. These findings inform about these models' interpretability and optimal adaptation for downstream tasks.

Together, these contributions present an understanding of vision in terms of active search, extraction and binding of information. Such an active search flexibly allows the usage of individual elements along with the relationship structures of those elements in the scene. By discussing the role of search and nature of representations in human vision during scene understanding, we explore the computational inferences and algorithms that can be adapted for robust machine vision.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Acronyms

BLIP Bootstrapping Language-Image Pre-training. 22

CLIP Contrastive Language-Image Pretraining. 21, 32, 34, 59, 61
CLS Classify. 22
CNN Convolutional Neural Network. 17, 21, 22, 26, 27, 43
COCO Common Objects in Context. 18

DINO Self-Distillation with No labels. 22, 23
DL Deep Learning. 8, 21, 25–27, 30, 49, 64

GAN Generative Adversarial Network. 24, 35, 47

SAM Segment Anything Model. 22, 23, 62

VAE Variational Auto-Encoders. 24
ViT Vision Transformer. 21, 22
VLM Vision Language Model. 21, 23, 26, 27, 49, 61, 62
VQA Visual Question Answer. 25, 26

LIST OF ORIGINAL PUBLICATIONS

[I] **Tarun Khajuria**, Kadi Tulver, Taavi Luik, Jaan Aru. Constellations: A novel dataset for studying iterative inference in humans and AI. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* 2022 (pp. 5142-5152).
https://doi.org/10.1109/CVPRW56347.2022.00562
Author contributions: methodology, implementation, analysis and writing manuscript

[II] **Tarun Khajuria**, Kadi Tulver, Jaan Aru. Comparing a computational model of visual problem solving with human vision on a difficult vision task. In: *PLOS Computational Biology*, December 2025.21(12):e1012968
https://doi.org/10.1371/journal.pcbi.1012968
Author contributions: conceptualisation, methodology (computational models), implementation (computational models), analysis and writing manuscript

[III] **Tarun Khajuria**, Braian Olmiro Dias, Marharyta Domnich, Jaan Aru. Interpreting the structure of multi-object representations in vision encoders. In *Proceedings of the World Conference on Explainable Artificial Intelligence*, 2025. vol 2577. (pp. 359-382). Springer, Cham.
https://doi.org/10.1007/978-3-032-08324-1_16
Author contributions: conceptualisation, methodology, implementation, analysis and writing manuscript

1. INTRODUCTION

Human vision is extremely adaptable and robust to changing conditions. We can rely on our vision to assist with a range of tasks, from driving to reading to playing badminton, and in diverse conditions, such as in a storm or a cloud of dust, under partial occlusion, and in various lighting conditions. In cognitive science, this ability is attributed to the generative nature of vision, where there is an interplay between bottom-up and top-down signals. Bottom-up signals refer to the raw sensory data that travels from the bottom of the system (the eyes) up to the brain. Conversely, top-down signals represent feedback from higher cognitive regions i.e. the parts of the brain responsible for memory and prior knowledge. The iterative interplay between these two kinds of signals allows us to identify and update the current context, then interpret the incoming visual inputs according to this context [1, 2]. This cycle of recurrent processing, wherein high-level predictions are constantly matched against low-level sensory data is a primary driver of the robustness and adaptability of human vision under uncertainty [3, 4].

Another important property of human vision is its ability to flexibly focus on various aspects of the visual scene based on the context [5]. This flexibility allows us to identify the overall structure of a scene, including high-level details like lighting and object arrangement [6]. Simultaneously, we can focus on a particular element if the task requires treating it independently. Switching between the part-whole hierarchy in the scene plays an important role in forming contextual understanding [7]. This process helps fill in gaps of missing sensory information and allows the observer to avoid acting upon spurious correlations [5].

The iterative interplay between these hierarchical representations can be modeled computationally through recurrent neural networks (RNNs). When dealing with challenging conditions such as heavy occlusion or visual clutter, recurrent models have demonstrated greater robustness than their feed-forward counterparts of equivalent depth and parameter count [8, 9]. Further, they can better explain human behavioural patterns such as the trade-off between speed and accuracy and better match the representation of the primate visual system in challenging visual tasks [4, 10, 11]. However, because these models are largely trained on discriminative objectives, optimizing only for the classification of inputs, they often fail to account for the generative components of human vision. These components refer to the brain's ability to perform internal synthesis and predictive reconstruction, where prior knowledge is used to actively render a mental hypothesis to match sensory data [12, 13]. Discriminative training typically lacks this reconstructive feedback loop, limiting the model's ability to resolve ambiguity through structural reasoning about the scene.

Analysis-by-synthesis is a computational framework that treats vision as a process of inverse graphics, where the system understands an image by attempting to reconstruct it from internal mental models [12, 14]. Unlike purely discriminative models, which map pixels directly to labels, analysis-by-synthesis algorithms use

a generative model to synthesize a hypothesis and then compare that hypothesis against the incoming sensory data. Multiple implementations of this framework have been proposed for complex vision tasks such as solving highly degraded Mooney images [15], and 3D face perception [14]. Analysis-by-synthesis algorithms demonstrate superior performance on complex images compared to standard Convolutional Neural Networks (CNNs) [15]. However, the high computational latency of current implementations remains a significant drawback; consequently, researchers propose integrating feed-forward components into both analysis and synthesis stages to increase operational efficiency [14].

In comparison to both recurrent neural networks and analysis-by-synthesis frameworks, stand-alone feed-forward computer vision algorithms are fast but not robust to perform well on out-of-distribution inputs and tasks [16, 17]. As shown by Geirhos et al. [18], standard feed-forward CNNs often over-rely on low-level features and textures to perform classification tasks. While such reliance often leads to high performance when inputs and tasks fall within the training distribution, these models can be prone to learning shortcuts in the form of spurious correlations [19]. Depending on the complexity of the data and the specific task, such correlations may not generalize to inputs that are structurally dissimilar to the training set, potentially resulting in a vision system that is less robust than its biological counterpart [20].

In this thesis, we further explore the differences in computational mechanisms between human and machine vision for the representation and understanding of the scene. In chapter 3, we first design and propose a complex vision task inspired by star constellations, where humans and machine algorithms try to find an object hidden in the image with little signal defining the object [21]. We show that the process of solving these images in humans involves the active generation of hypotheses and iteration over multiple versions of a single hypothesis or many hypotheses while solving the task.

In chapter 4, we propose a generative search solution, termed GenSearch, to solve constellation images containing digits and objects from the MNIST and Fashion MNIST datasets [22]. This algorithm operates by searching within the latent space of a pre-trained image generator to identify the specific image whose outline best fits the provided constellation dots. The algorithm searches in the latent space of an image generator trained for these datasets and tries to find the image whose outline fits the constellation dots the best. We further compare this algorithm to humans solving the same images and find several interesting similarities in their behavior, including the mistakes made, the maintenance of multiple hypotheses and confusion between similar objects. We further evaluated the task using standard feed-forward Convolutional Neural Network (CNN) (such as ResNet-18) trained specifically on these constellation images. These networks achieved near-perfect classification accuracy on the training distribution, significantly exceeding the performance of both human observers and GenSearch. This distinction becomes evident when examining how classification accuracy changes

across varying difficulty levels, defined by the number of dots available to represent each object. As the number of signal dots decreases and task difficulty increases, the performance of human observers and the GenSearch algorithm degrades at a similar rate. In contrast, the trained CNN models remain over-fit to the specific dot densities encountered during training and fail to generalize when the sparsity of the constellation changes. These results suggest that the generative search approach better captures the robust and adaptive nature of human visual reasoning.

Finally, in chapter 5, we test some of the popularly used vision encoders in deep learning on natural scenes with multiple objects using the Common Objects in Context (COCO) dataset [23]. We designed a two-object representation probing task to analyse the representations of these vision encoders at various layers and token types. The task quantifies the decodability of two objects in a multi-object scene from a given embedding. The primary goal of this task is to quantify representational interference, the phenomenon where the features of a dominant object overwrite or suppress the features of a less salient object in the model's fixed-length embedding. We find that irrespective of the training objective or the architecture, certain prominent objects are better represented than others in the later layers of these networks. In contrast, many background objects might not be well represented in the token used for the downstream tasks. The inferences from this study highlight the need to re-examine the use of output tokens from static pre-trained vision encoders in downstream tasks and other general multi-modal models.

Overall, in this thesis, we present an overview of the understanding of scene understanding in terms of active usage of information, such as in the part-whole object hierarchy. Where the information from the scene is actively accessed multiple times during the inference process as compared to a single feed forward pass through the inputs. Such active search allows the usage of individual elements in the scene along with the relationship structures in which those elements are likely to co-exist in the visual context. The specific contributions of this thesis aim to characterize how visual scenes are represented and reasoned about both as a whole and a collection of parts, in humans and machine algorithms. Drawing on the three contributions, we then discuss the role of abstract visual representations and active search in supporting visual perception during scene understanding.

2. BACKGROUND

This chapter provides an overview of topics on the intersection between biological vision and computational modeling. The first section examines the complexity of the human visual system, focusing on active search and the hierarchical nature of scene representation. Subsequent sections discuss the evolution of computer vision architectures, from early convolutional neural networks to modern vision transformers and generative models. By synthesizing these diverse fields, this chapter establishes the theoretical framework necessary to understand the analysis-by-synthesis paradigm and the specific experimental probes used in this thesis.

2.1. Vision and its complexity in humans

In humans, vision is not performed in isolation but rather involves various aspects where existing knowledge and awareness of the current surroundings play an important dual role in understanding the scene. On the one hand, input from the scene is used to extract relevant information about the environment[1]. However, the human visual system not only extracts information from the incoming visual inputs but also decides, based on the current context, the task at hand and previous experience, where to direct the gaze next to collect more information [24]. Further, within the information being received by the eyes, humans can focus on a certain part of the scene to get more detailed input. Also within the objects modelled, we can look for details of the objects or can focus on the relation of the objects in the scene [25]. Such sophistication in human vision allows humans to operate under various conditions and dynamically extract information for complex tasks.

2.1.1. Gestalt Psychology

A central function of vision is to abstract information in a way that makes sense to us. This form of abstraction in human vision that occurs at a scene level has been partly captured by principles of Gestalt psychology [26]. These principles are often summarized with the well-known phrase that 'The whole is something else than the sum of its parts'. In relation to scene structure learning and abstraction, these principles identify how human perception can be affected by the aggregate information in the scene while abstracting away individual details. The Gestalt principles also describe the grouping effect of elements based on certain common properties in the scene, such as proximity, similarity, continuity and common fate.

Gestalt principles are directly relevant to this thesis as they provide the psychological basis for the part-whole hierarchy discussed in later chapters. Specifically, the grouping effects based on common properties such as proximity, similarity, and continuity allow the human visual system to resolve ambiguity in sparse or noisy environments. This thesis investigates how such grouping mechanisms can be modeled computationally, particularly in Chapter 3, where we examine how

humans utilize Gestalt-like "hypotheses" to identify objects hidden within sparse constellation dots. By understanding these biological grouping rules, we can better evaluate why standard machine vision models often fail to perceive the "whole" when local signals are degraded.

2.1.2. Vision and Search

While Gestalt principles describe how the visual system organises static information, the human observer rarely perceives a scene in a purely passive manner. Instead, the brain must actively navigate the visual environment to locate specific information, a process known as visual search. More specifically, visual search is the task of looking for a target object in a scene cluttered with other objects called distractors. We perform this task ubiquitously, while searching for a particular object or looking for content of interest in a scene, or directing our gaze at every instance towards the key points in an environment. Visual search entails both active goal-driven search and passive attention shifts towards the most interesting elements of a scene. The mechanisms behind these different types of search form a critical part of vision and are an example of the active observe-act-observe cycle in vision to collect information about a scenario [24].

One of the most prominent theories explaining visual search is the feature integration theory [27], which explains that certain features are explicitly encoded by the visual system, and hence, upon presentation of a target that is distinguished from the distractors by such features, the target pops out in the visual scene. On the other hand, more complex features need to be bound and processed at the object level. As a result, the reaction time in a search where the target differs from distractors by more complex features or a conjunction of features is a direct function of the number of distractors. Many newer theories such as guided search theory [28] considered the increased reaction time in conjunction search due to a lack of pre-attentive guidance from all features under resource constraint. Newer versions of guidance theory [29] take into account many factors of pre-attentive information affecting the search such as 1. top-down (target or goal), 2. bottom-up (prominent features in scene), 3. prior history or priming, 4. reward, 5. scene syntax and semantics.

2.1.3. Scene understanding in human vision

While the observe-act-observe cycle describes the behavior of visual search, the physical implementation of these processes occurs across two specialized neural pathways in the brain. The ventral stream is typically associated with object identification and categorisation. It represents object category-related information in an abstract form and hierarchically encodes the part-whole relationships that lead to the object category [30, 31]. The dorsal stream represents other aspects of the visual scene related to action affordances of the objects [32, 25], particularly representing the relationships between objects in the scene in an abstract form,

i.e., without attaching object categories to these representations [32, 33]. Such relationships can be scene-level aggregates, i.e., the gist of the scene or abstract relationships between objects that can still be in the form of retinotopic maps, for example, representing object salience or that the object on the left is bigger than the object on the right, etc.

Together, the visual system processes and maintains various aspects of the scene. It is considered that flexible access and the dynamic combination of these different types of information allows the visual system to analyse scenes in a robust manner [34, 35].

2.2. Computer vision

Vision-based tasks have been a major driver of the advances and adaptation of Artificial Neural Networks. From the proposal for the first CNN architectures by Fukushima [36] to the first time DL algorithms started showing impressive results in the ImageNet classification challenge [37]. CNNs became the standard architecture to encode images and were used in various forms and in conjunction with various other layers for tasks such as image segmentation [38], image captioning [39], etc.

2.2.1. Vision encoder

A vision encoder is a trained neural network designed to transform raw pixel data into a structured internal format known as a representation. One of the key features of modern DL is the ability to transfer learnt representations from one task setting to another [40]. This key property makes the use of encoders popular in the field. One such example is CNN backbones trained on image classification that have been reused and transferred to other tasks directly or by fine-tuning. CNNs trained on ImageNet quickly became a standard starting backbone for computer vision projects and were used for feature extraction for tasks as different as image segmentation [38]. This use of representations from a pre-trained network for the downstream task has also led to a wide variety of research in learning a good representation for the image inputs.

A good representation is one that contains the correct details for the downstream task [41]. Vision encoders are now trained using various objectives to obtain this general representation suitable for a variety of downstream tasks. Self-supervised learning objectives played an important role to advance this field as they allowed the networks to be trained on larger quantities of image data without any explicit need for labels [42, 43]. Furthermore, the use of image encoders that were trained along with a language task provided better semantic grounding to the image encodings [44]. With the popularising of Vision Transformer (ViT) as image backbone, vision encoders started to be scaled to larger models that were trained on a very large amount of data. Some of the key types of such vision encoders are ViT trained on image classification [45], Vision Language Model (VLM) like Contrastive

Language-Image Pretraining (CLIP) [46] and Bootstrapping Language-Image Pre-training (BLIP)[47] trained on image-text contrastive learning and captioning, self-supervised models such as Self-Distillation with No labels (DINO) [48, 49] and image segmentation models such as Segment Anything Model (SAM) [50].

2.2.2. Neural Architecture for vision encoders

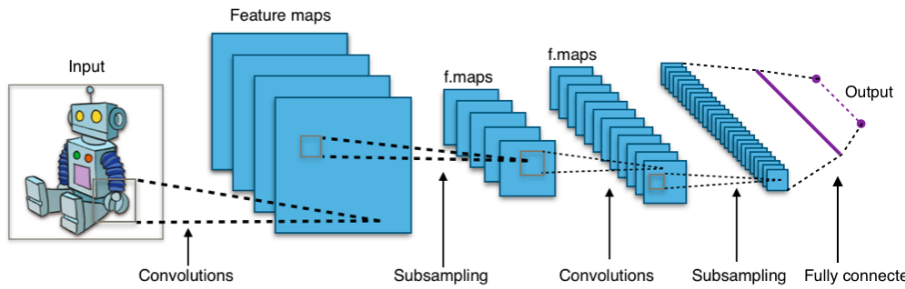


Figure 1: A CNN network showing its two main layer operations, convolution and pooling during an image classification task [51]

CNN architecture as shown in Fig 1 is primarily based on the use of a convolution operation between visual signal and kernels (filters). The visual signal, typically in the form $H \times W \times C$ (where H,W,C are the height, width and number of channels of input signal) is convolved with filters, typically in the form $N \times N \times D$, where in many cases a small value of N is preferred (typically 3 is used) and D is chosen according to the number of channels of input signal. In the CNN networks, the filters are typically initialised with random values and then learnt during the training of the network. These filters learn to extract some particular features from the part of the image signal they are applied. As more CNN layers are stacked on each other, the filters have increasingly larger receptive fields of the input image, allowing them to analyse a larger area of the input image. Another important operation used in the CNNs is the pooling layer, which is used to subsample the signal in the H and W dimensions. This allows for lower dimensionality in the later layers as well as a rapid increase in receptive fields for the later filters.

ViT [45] is another architecture popularly used in image encoders. This architecture is the adaptation of transformers [53] which were originally designed for text and sequence encoding to an image classification task. Figure 2 illustrates how the input image is cut into patches and linearly encoded into the transformer inputs as tokens. A positional encoding is added to this representation that represents the position of the patch in the image. The learnable operation in the transformer is the self-attention between the tokens, learning how tokens should transform their representations based on the representation of other tokens in that layer. To train the model, an additional token called Classify (CLS) token is used in addition to the image patch tokens. The final representation of the CLS token is used to obtain the classification category and calculate the loss to train the network.

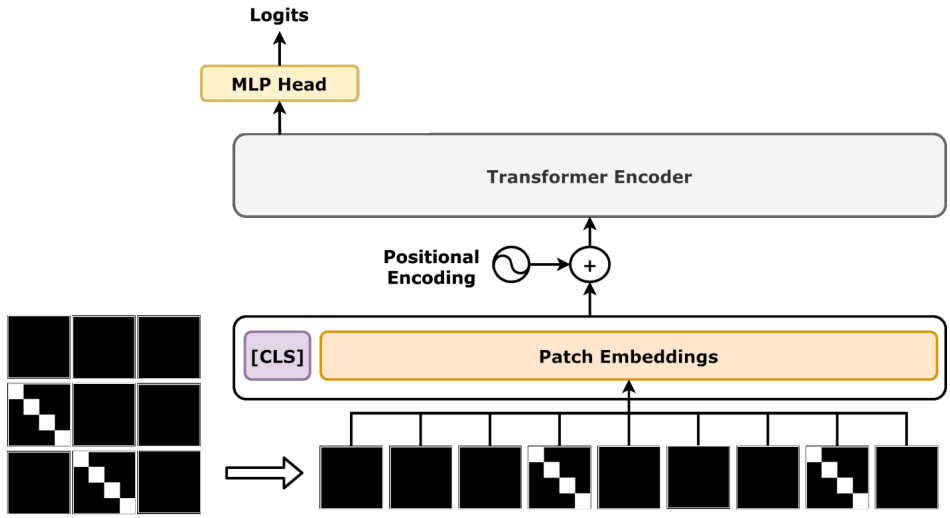


Figure 2: A vision transformer model, which inputs images by dividing them into small patches which are processed through the model as individual tokens. A special CLS token is appended that predicts the final class label. [52]

2.2.3. Structure representation in vision models

VLMs due to their alignment with text are some of the most useful models for downstream tasks, as many tasks can be defined in a zero-shot manner with the use of images and relevant text descriptions. While these models are functionally very useful, many studies have pointed out the inability of these models to represent the structure of the scene, showing poor performance on tasks requiring knowledge about the space and the instantiation and arrangement of the objects in the space [54, 55]. Tasks such as counting, instance segmentation, or object localisation are better handled by models explicitly trained on them, such as SAM. These models, on the other hand, do not represent the high-level semantic meaning between object structures. The DINO family of models which are trained to capture the invariance of object representations under shifted viewpoints, offers a compromise by capturing space related properties of the scene, its objects and its parts, while also modelling some high-level semantic associations between different objects and their components [48, 56].

2.2.4. Generative image models

All the models we have so far discussed are discriminative in nature, as they try to capture the representation of visual data by learning to align or separate the representations of different points. Only models such as BLIP, with its image captioning output, try to learn the generative description of the scene using language. While these models are easier to train and work well on specific tasks, a more general model is expected to infer the properties that generate the data itself. Hence, in this section, we discuss a few generative vision models relevant to this thesis.

Variational Auto-Encoders (VAE)s [57] are a model family that first encode the image input into a set of low-dimensional code and then sample from this code to try to re-generate the same image. VAEs introduce a probabilistic component to ensure that the latent space is smooth and continuous. In this way, the model learns this low-dimensional code and arranges it in a way that captures some of the semantic and structural regularities of the training image data. While these models could capture many of the latent features that explain the generation of its training data, the quality of images generated by the VAEs is suboptimal [58].

Generative Adversarial Network (GAN)s[58] are another class of models that learn to generate an image by matching the distribution of the image samples through a competition between a generator network and a discriminator network. In GANs (see Fig3), both the networks are randomly initialised. Then, during training, the generator network (G) samples a latent vector z and generates an image based on its current weights. The discriminator network (D) has to discriminate between the images generated by the generator and the real images from the dataset. The loss of the discriminator penalises it for incorrectly labelling the images as belonging to the generated or original category. On the other hand, the generator's loss penalises it if the discriminator can identify the difference between its generated images and the original. The networks are both used and their weights are updated in turns, creating a competition between the networks, which leads to their training. In case of successful training, the generator learns to generate images that resemble the distribution of images in the original dataset given a latent vector z. The images generated by these models are generally of higher quality than those of VAEs. The latent space of GAN's generator also preserves some structural and semantic relations in the image space, allowing meaningful steering operations to be defined between points in the space.

Diffusion denoising models [60] are a prominent class of generative models capable of producing high-quality images. These models operate in two phases. The first is the forward diffusion phase, where Gaussian noise is added to the image across T timesteps. At the end of this phase, the image is transformed into isotropic Gaussian noise. This phase is essential because it allows the model to learn the reverse mapping; by seeing how an image is destroyed, the network learns the statistical distribution of the noise it must eventually remove. In the reverse diffusion phase, a neural network (typically a U-Net) is trained to take the noisy image x_t and the current timestep t to estimate the noise present, thereby allowing the iterative recovery of the image from t to $t-1$. Later iterations, such as Stable Diffusion [61], perform this process within a compressed latent space rather than pixel space and incorporate cross-attention mechanisms to condition the generation on text inputs.

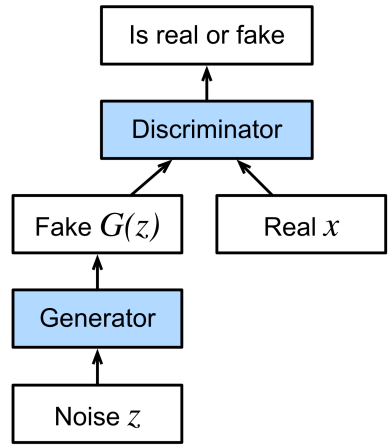


Figure 3: A schematic of a Generative Adversarial Network, where the generator learns to generate images based on the latent vector z and the discriminator has to identify the difference between this generated image and the samples from the real dataset. The model is trained through the competition between the generator and the discriminator, where the generator always tries to generate images that can fool the discriminator. [59]

2.2.5. Visual Reasoning

Visual Question Answer (VQA) as a task evaluates a model's capability to answer questions about an image, requiring complex visual and linguistic reasoning. The field was started by benchmarks like VQA [62], which proposed an evaluation using open-ended, free-form questions and expected solutions in natural language. Subsequent datasets, such as Visual Genome [63], provided content-rich images with explicit annotation of object positions and relationships, promoting models that could exploit structured information. Critically, datasets like CLEVR [64] were specifically designed to correct for shortcuts that deep learning models would exploit in existing benchmarks, forcing models to rely on explicit compositional reasoning rather than superficial correlations.

This requirement for explicit compositional visual reasoning i.e the ability to combine and manipulate discrete visual concepts (e.g., "the small blue sphere to the left of the large red cube"), exposes a major challenge with generalization in DL methods. This challenge has been comprehensively discussed [65] in light of the critical need for explicit inductive biases that allow for discrete yet flexible information binding of visual attributes and objects. Discrete symbolic representations, which facilitate the structured manipulation of concepts, are consequently considered a prerequisite for robust compositionality and reasoning in VQA systems [66]. While some new studies propose that such discrete information binding may emerge organically from training existing models on large datasets [66, 67], many other works advocate for explicit architectural solutions, such as discrete bottlenecks [68, 69], to impose structure on the latent representations used for

visual interpretation.

As per the requirements of this task, which demands both visual understanding and natural language output, the models designed or used to solve VQA are typically VLMs that integrate a language generation component capable of processing both vision and language inputs. Reasoning in VLMs is a key area of study, specifically how they attend to the image and utilize extra reasoning sequences during the language generation phase. Current VLMs struggle to ground their reasoning on the visual inputs and rely on the language processing and reasoning to answer the questions [70, 71]. To improve the faithfulness of generated answers and mitigate hallucination, recent work [72] has explored mechanisms like visual look-backs. These methods have attempted with some success to make the VLMs explicitly re-attend to the input image representations during the generation of the natural language response, which in turn improves its visual grounding and response accuracy.

2.2.6. Analysis-by-synthesis

In a purely visual reasoning setting, the analysis-by-synthesis paradigm tries to capture the iterative generative hypothesis testing used in difficult vision problems. The search for candidates in the generative solution shape makes this approach akin to visual reasoning, closely resembling reasoning described in the previous section for vision-language models. Yuille and Kersten [12] re-emphasised analysis-by-synthesis, i.e., using a top-down signal to guide bottom-up search under Bayesian inference as a prime candidate to explain real-world complexities in scenes. In processing highly degraded images, analysis-by-synthesis has been a primary candidate algorithm used to explain human performance as it was found to be a better match to human performance in identifying highly degraded Mooney images compared to conventional feed-forward deep learning algorithms [15]. Yildirim and colleagues [14] discussed the current analysis-by-synthesis algorithm's inability to scale to many real-world problems and being too slow to explain human vision. They further proposed using feed-forward modules as part of the algorithm to make the search more efficient.

Increasingly, these algorithms are being applied to more natural scenes, which has led to more hybrid implementations with conventional deep learning architectures. In images of natural scenes, [73] introduced parsing the scene in multiple iterations to model the components and their relations and create a scene graph. Ullman and colleagues [74] proposed a solution to scene analysis using a goal-guided algorithm that uses a CNN to extract bottom-up features and then produce a top-down command token that guides the re-extraction of other bottom-up features. The system iterates between using the bottom-up and top-down network to focus step-by-step on relevant parts of the image and creates a scene graph as required by the goal of the task. Further, [75] used similar insights from analysis-by-synthesis to inspire changes to attention mechanisms in modern DL architectures, which

improved performance across tasks incorporating this context-driven top-down attention.

2.3. Alignment between computer vision models and human vision

The discovery of receptive fields in the visual cortex by Hubel and Wiesel led to the understanding of the hierarchical nature of representation in human vision [30]. With the CNN based models solving the image classification task, the mapping of the representations of various layers of the CNN model to the areas along the human ventral stream provided a useful model of object recognition [76, 77]. However, as described above, the neuro-circuitry of vision is typically considered to happen in two streams, where the ventral stream is particularly involved in deciphering the contents of the visual scene, following the hierarchical feature discrimination leading to semantic categories. On the other hand, the dorsal stream is known to aggregate scene statistics and maintain scene gist, guiding inference of the structure of a scene, which includes understanding the position of the objects, counting etc. Even though few deep learning models are trained specifically for tasks attributed to the dorsal stream in humans [38, 78], general deep learning (DL) models such as VLMs are known to perform badly on scene structure-related tasks such as counting, searching and spatial reasoning tasks [79, 55, 54]. It is an active research question in both DL and cognitive sciences to create models of how these two streams of visual information processing work with each other to obtain robust visual inference akin to human vision [80, 81, 82]. This emphasises the role of understanding these functions in natural vision and creating computational models that can mimic the operations of natural counterparts.

2.3.1. Probing of representations

Probing is a methodology used in the analysis of both brain and DL model representations [83]. These concepts can further be aligned between humans and computer vision models. Probing involves learning a small model using a set of representations produced by a region of an information system in response to a stimulus. The model tries to learn a fit (typically a 'linear' fit) between the representations and some labelled property of the stimuli, such as colour, numerosity etc. The model's performance on a hold-out set is used as the criterion to evaluate how well the particular representation region encodes for the tested property of the stimuli. It is important to limit the probe side to small models to ensure that the accuracy of the probe shows the goodness of the representation in encoding the tested property, as opposed to the learning ability of probe [83, 84]. A more complex probe can even learn to model very complex relationships between the representations and the property, given enough model capacity and training samples. We will use probing in our experiments to understand how the concepts are represented in popular DL based vision encoders.

3. HOW TO STUDY ITERATIVE INFERENCE IN HUMAN AND MACHINE VISION

The central challenge in modern computer vision is the disparity in robustness between biological and artificial systems. Here, robustness refers to satisfactory performance over a range of tasks in a variety of conditions. While dynamic tasks like autonomous driving or robotic manipulation highlight these disparities, investigating them in such environments is computationally and financially costly due to the need for real-time physics simulation and the complexity of behavioral tracking in a live task. Furthermore, the high-dimensional nature of dynamic tasks makes it difficult to isolate whether a failure occurs in the visual perception system or the motor-control logic.

To bridge this gap, we propose a minimalist, static image-understanding task that isolates the perceptual mechanisms required for scene resolution. This task promotes the use of iterative inference, which is defined as the process of dynamically generating, testing, and refining internal generative hypotheses to resolve sensory ambiguity [85].

When trying to find star constellations in the night sky, one cannot immediately see objects formed from stars, but can still generate many hypotheses about what could be there (“a bathtub”, “a boat” or in the absence of any complex shape we fall back to finding simple shapes like ’lines and curves and circles’). One can also test whether these hypotheses explain patterns of stars in the night sky or not in an iterative fashion. Eventually, one might discover a warrior (Orion) or a bear (Ursa Major) as a solution. This process of going back and forth, revising the hypotheses until a solution is found, has been called ‘iterative inference’ [85].

This chapter presents a comprehensive framework for studying these mechanisms through three primary contributions. First, we detail the construction of the Constellation Dataset, which obscures common objects from the THINGS, MNIST, Fashion MNIST, and Sketch datasets within sparse, noisy dot patterns. Second, we describe a generation pipeline that allows for the precise manipulation of signal-to-noise ratios to control task difficulty. Finally, we report the key observations from human pilot experiments alongside a baseline evaluation of the CLIP model, inferring that while humans successfully utilize iterative strategies to resolve these images, standard feed-forward architectures fail to generalize to such sparse modalities.

The code for constellation image generation is available at: `https://github.com/tarunkhajuria42/Constellations-Dataset`

3.1. Constellation image generation

The objective of the generation pipeline is to transform natural images into "constellations" that preserve global geometry while eliminating local texture. This

specific design is chosen because it prevents feed-forward models from relying on local feature correlations, thereby forcing the visual system to engage in iterative, structural reasoning. While many datasets provide existing segmentation masks, we utilized a custom pipeline to use across diverse sources, including those without pre-existing ground truth.

The steps to generate the constellation images are illustrated in Figure 4 and are the following: 1) Generating outlines for the object from the original image; 2) Manual selection of the best outline candidate, followed by manual editing in some cases; 3) Automated generation of the dotted version and then the constellation version of the image using the selected outline. Next, we discuss these steps in detail.

1. First, we use Mask R-CNN [38] to identify the region of the image containing the object. We obtain the binary mask output from Mask R-CNN, indicating the pixels belonging to that object in the image. We multiply this binary mask with the original image to get the image with only the detected object. In particular, we use Mask R-CNN pre-trained on the COCO dataset [86], which contains many categories overlapping with the objects in the Things dataset.
2. Second, to compensate for the inconsistency from Mask R-CNN outputs, we repeated step 1 with multiple mask settings to obtain multiple masked images. We generated multiple masked versions by capturing 1) only the principal object, 2) the first two prominent objects, 3) an unmasked full image. Having these multiple versions allows for a simple manual selection later, making the final outputs more appropriate.
3. We then use the Canny edge detector [87] to obtain the outlines from these images. The threshold of (100,200) is used for all masked images. For the unmasked images, an additional blurring mask with a radius of 5 pixels is used. The image obtained after these steps can be seen in step 2 of Fig 4.
4. In this step, we generate an image representing the outlines with dispersed dots. These dots are separated by a regular pixel distance ‘d’. We traverse the image pixel by pixel to find any white pixel on the outline image and draw a black circle around the pixel with radius ’d’. Finally, the radius of the white points can be increased to adjust visibility.
5. We add noise to the generated dotted image, traversing it again and adding dots to each pixel with a uniform random probability of ’p’

3.2. Datasets generated and their features

We released the constellations datasets to increase its relevance for various research communities. The primary release used the THINGS dataset [88] to create a version of the constellation set. The THINGS dataset consists of 2-3 examples of everyday objects with a total of 1215 objects represented in 3533 images. The objects in the

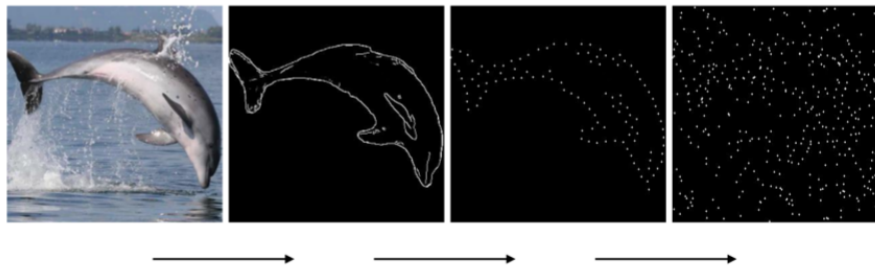


Figure 4: Constellation image generation process: 1) We use Mask-RCNN to find the object mask and then use canny edge detector to obtain the object outline. 2) We leave the dots at equal distance 'd' by drawing black circles. 3) We add additional distractor dots at each pixel with a uniform probability 'p'.

dataset consist of everyday objects such as animals, kitchenware, furniture, clothes, food, etc. This image set is important for the cognitive science community, with this set being a part of the THINGS Initiative [89], where stimulus variations and various human experimental data are shared on the same set of images. This can be useful for further studies that can cross-reference data and insights to make larger connections and gain stronger experimental evidence.

The constellation set for THINGS dataset has only 2-3 examples per category, which can be used to test machine learning algorithm's few-shot performance. However, to train and benchmark existing machine learning techniques on this type of stimuli, we created larger constellation sets where more examples per category are present. In this spirit, we first generated and released the constellation version of a popular sketch dataset [90], consisting of 20,000 images covering 250 objects. When recording humans solving the constellation task one way to record their attempts is by asking them to sketch their solutions. Hence the constellation solving task can be of interest to the sketching in DL community. As this dataset with its human strokes data is of interest to the deep learning sketching community, we found it fit to create a constellation version of this dataset to attract the interest of the skecting in DL community. Finally, we also released constellation versions of two popular machine learning datasets, i.e, MNIST and Fashion MNIST, which consist of a large number of training and test images with 60,000 and 10,000 images, respectively. These sets have only 10 categories each and have already been used as baseline examples in a variety of machine learning and deep learning work.

We generated multiple versions of the constellation images with different distance between the dots 'd'. We can see in Fig 6 how changing 'd' changes the difficulty level of the images. These variations allow the difficulty level of constellations to be adjusted to best induce the difficulty for iterative inference. In the machine learning experiments, this difficulty level change allows for testing for controlled distribution shifts under the same constellation image modality.

Figure 5: Six examples of the constellation objects. Depicted are the original image, the dotted outline and the constellation image with low local signal. The constellation dataset allows researchers to train and evaluate sketching or other generative solutions for inferring the object hidden in dots.

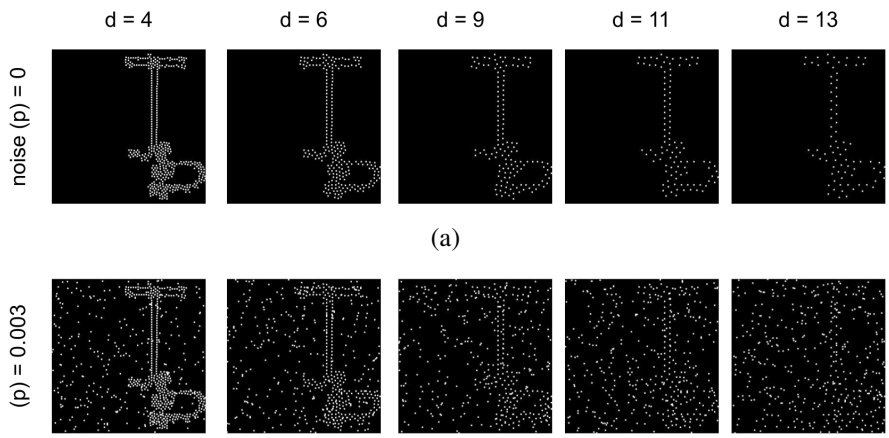


Figure 6: Dotted and constellation images with different difficulty levels. A combination of 'd' and 'p' is chosen to get the optimal difficulty level for images to be used in experiments. Fig. 6a: Dotted version with different distance 'd' between dots. Fig. 6b: The respective constellation images with noise (p = 0.003).

3.3. Humans follow iterative inference while solving constellations

To examine the dynamics of iterative inference, we conducted a pilot study where participants were permitted to view the constellation images for an unrestricted duration. The primary task was to identify the hidden object and physically externalize the perceived structure using a touchscreen monitor and a stylus. To gather more information about the process of solving the image, we also recorded the participant's voice as they explain their thought process while solving the task.

These participants, when prompted to find objects in constellation images, reported iteratively refining their candidate solutions based on their initial guess. For instance, when a constellation image hiding a hairdryer was presented, the participant found the handle part of the contour but assumed that it could be a leg and tried to then fit a cat to the constellation's shape. But a not-so-good fit prompted a re-evaluation of this feature, where it was next tested for being a wing of an airplane and finally recognised as a handle of a hairdryer.

The strategies and the search dynamics varied across participants and the image difficulty. For many images, participants reported finding the shape in the first instance, whereas for more difficult images, they reported that in the first instance only a few simple lines and curves popped out which then led to further use of these candidate curves to be composed using the shape of some higher features. The overall strategy used by the participants can be summarised under the umbrella of iterative inference, where the participants iteratively collected cues to generate and then test those hypotheses.

3.4. Pre-trained model CLIP's performance on the constellation dataset

To establish a baseline for modern computer vision, we evaluated the zero-shot performance of CLIP (Contrastive Language-Image Pre-training) [44].CLIP as a pre-trained model can be used for zero-shot image classification over different modalities. It has also been used to semantically guide image generation based on textual descriptions. As the model has been used in many deep learning based sketching projects [91, 92], we found it relevant to check the performance of variants of CLIP on various modalities leading up to the constellation images.

We evaluated four variants of CLIP (Vit B/32, Vit B/16, Resnet 50X16, Resnet 50X4) on various modalities of our dataset images by setting up a classification task based on the categories provided with the Things dataset [88]. We found that the pre-trained model's performance drops drastically from the original image to the constellation image (see Figure 7). Performance with constellation images was at near random guessing levels.

For this experiment, all images for each of the modalities (Original, Outline, Dotted, Constellations) were collected from one of the main dataset folders (p =

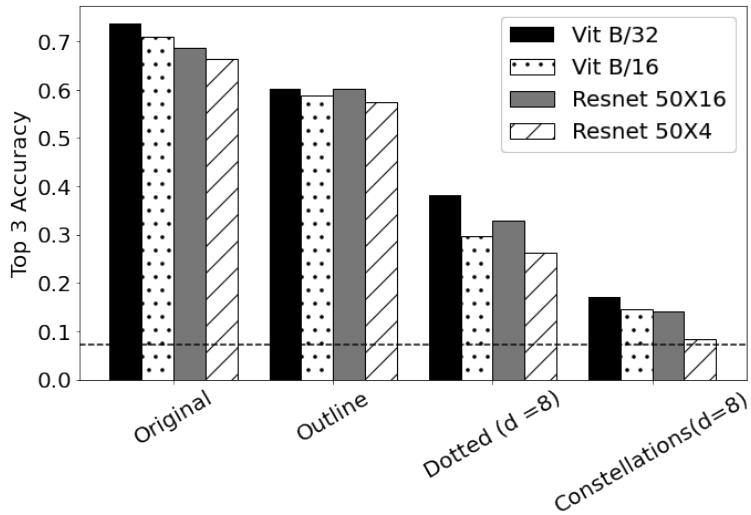


Figure 7: Model accuracy for CLIP versions Vit B/32,Vit B/16, Resnet 50X16, Resnet 50X4 models on different modalities of images in the constellation dataset. Dotted line shows the baseline top 3 accuracy for random prediction.

0.002 or p=0.003) into separate folders. The class label for each image was obtained from the 'Top-down Category (manual selection)' field in things_concepts.tsv in folder main/ from https://osf.io/jum2f/ (Things dataset). As the category is not provided in this field for each image, only images with a given category in this field were selected for this evaluation. Hence we used a total of 2566 images in each modality during this evaluation (The full dataset has 3533 image sets). Where more than 1 label was provided for an image, all labels were considered correct while evaluating the classification performance. Considering all these details, we were left with 41 labels in the final classification task. We reported a top-3 class accuracy i.e. considered the classification to be successful if one of the top 3 predictions of the model is a correct label.

3.5. Discussion and main takeaways

In this chapter, we presented a new dataset, 'constellations', inspired by star constellations. Using this dataset in our pilot human experiments, we demonstrated that human participants can be observed using iterative inference while solving these images. Participants were instructed to sketch their solutions and explain their thoughts during the process of solving, helping us to record and analyse the solving process. We provided the code and detailed the pipeline to create this dataset and released the constellation version of THINGS [88], MNIST [93], Fashion MNIST [94] and Sketch dataset [90].

Constellation dataset provides a unique experimental setup to capture the externalized internal search and test dynamics of vision, a feature absent in other

hard benchmarks. While datasets like Mooney images [15] and Contour Integration tasks [95] effectively challenge global integration, they are primarily used to measure final recognition accuracy or detection thresholds. While the constellation task with its ability to capture drawing response during the solving process allows to capture detailed features of the iterative inference process and final the solutions in form of the drawings.

We speculate that this stimulus type promotes iterative and generative solutions by minimizing the availability of local information. Local shape cues were deliberately removed, and additional noise dots were added to further obscure local structures. As a result, inference must rely on collecting information distributed across scales rather than direct bottom-up composition of local features. Iterative hypothesis testing thus becomes essential for solving the images, contrasting with the feed-forward processing of most conventional ANNs. We found that the zero-shot classification performance of CLIP on these tasks is quite low (see Figure 7). This suggests that direct inference of labels, or even meaningful initial conditioning for search guided by CLIP, is challenging when applied to dotted or constellation versions of object images.

The impact of this particular work is that it introduced a new experimental paradigm and dataset for complex image understanding, specifically designed to reveal the iterative nature of visual processing. This dataset and task will hopefully promote further experiments to better understand iterative processing in humans and computer vision systems.

4. GENSEARCH: GENERATIVE SEARCH WITH EVOLUTIONARY ALGORITHM

While most modern vision models rely on a single feed-forward pass, recent evidence suggests that generative classifiers possess intriguing properties regarding robustness to out-of-distribution noise and adversarial perturbations [96]. The task of categorizing objects from sparse or degraded visual cues—where local signal is insufficient for purely bottom-up recognition—is often hypothesized to involve an internal generative component or higher-level perceptual priors [12]. These mechanisms may allow the visual system to resolve sensory ambiguity by synthesizing expected object structures to "fill in" missing information [14]. We define solving of constellation images as correct identification of the object in the constellation image by drawing the correct contour on the image. As indicated in the pilot study presented in the last chapter, humans can utilise an iterative refinement approach to generate new solutions. Inspired by observations from human experiments, we developed a Generative Search algorithm (GenSearch). This chapter describes the GenSearch algorithm and evaluates how it compares to humans in solving constellation images.

The GenSearch algorithm follows a line of work in solving problems in vision under the umbrella concept of analysis-by-synthesis. This general framework has existed for many decades, but in 2006, Yullie et. al brought back focus to this framework and argued that many realistic problems can be solved by it [12]. Many works have used this idea for solving datasets such as Mooney images, 3D face perception and other tasks with ambiguous input signals [15, 14]. We are particularly interested in using the constellations dataset we created, where the level of difficulty can be controlled to find a sweet spot where visual perception becomes a task of active problem solving rather than simple recognition. In actual star constellations, there is no signal, but still, humans can often see a shape. In the case of our artificial constellations, this signal can be controlled to better bring out the generative problem-solving aspects of vision during extremely low signal inputs, while still allowing humans to find the correct solution.

The GenSearch algorithm uses a GAN based image generator to generate candidate images and tries to fit them on the constellation image by optimising for maximum count of dots the generated sketch passes through. The optimisation is performed using an evolutionary search. The overall algorithm is explicitly generative as it uses an image generator to generate candidate solutions. The search algorithm refines the solutions to find a perfect fit based on low-level cues in the constellation image. These low-level cues are the fitness of the shape to the dots in the constellations, evaluated by the number of dots the shape passes through. We observed that the solution found by the GenSearch algorithm matches humans in various aspects, such as accuracy, types of mistakes, and the solving process. Overall, we found that generative search, especially using search operations on

multiple hypotheses, can be a good candidate to explain many processes in human perception and problem-solving. The code and human-data used in this study is available at : `https://github.com/tarunkhajuria42/GenSearch`

4.1. Generative Search Algorithm

In this section, we explain the details of how the GenSearch algorithm is implemented. GenSearch consists of an evolutionary search in the latent space of a generator (GAN) guided by low-level cues from the constellation image. These low-level cues are evaluated by counting the number of dots passing through the edge of the candidate solutions and are used by the evolutionary algorithm as its fitness score. In Fig 8 one can see the steps involved in the GenSearch algorithm. The details for the steps are the following:

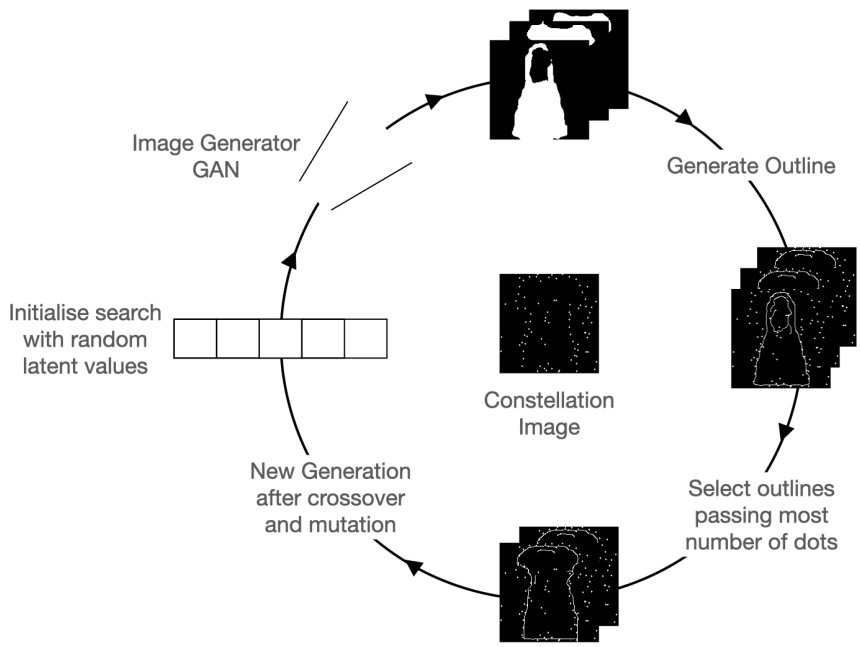


Figure 8: GenSearch Algorithm: The constellations image is solved by generating candidate solutions with a GAN and refined using a genetic search conditioned on best fitting the solution outlines to the dots on the constellation image.

1. The initial population of solutions is randomly sampled from an image generator's latent space. So we sample a set of vectors in the latent space of the GAN and then use them to generate candidate images with the GAN.
2. The fitness of these solutions is calculated by comparing the generated image's fit to the dots on the constellation image.
3. The calculation of an object's fit to the dots constitutes fitting the edges of the object contour to the dots. Hence, we first run an edge detection on

the sample's greyscale image and then use the obtained contours to check if they pass through a dot. We allow a tolerance of 3 pixels for accessing connectivity. Only dots connected by contour edges less than 40 pixels apart contribute to the fitness score, preventing the model from favouring excessively long, unrealistic drawings.

4. Finally, after the top 200 images from each population are selected based on their fitness, we generate the new population by point mutation (change of genes randomly) and crossover of the features (combination of genes from two parents, i.e., two of the top 200 candidate vectors) of these vectors. The process is repeated for 30 generations.
5. At the end of each generation, the solution with top fitness is considered the solution of that iteration (generation). The convergence is calculated based on the change in the fitness of top solutions over the generations.

4.2. Comparison of GenSearch and humans

We compare human and GenSearch's performance on two datasets: 1) MNIST constellations, and 2) Fashion MNIST constellations. Examples from both sets are shown in Fig 9a. Both sets have 10 categories each. In both cases, the train set consists of 60,000 images, and the test set consists of 10,000 images. Out of the 10,000 test images, we selected 38 images best suitable for human experiments, found fit to be at the optimal difficulty level and clarity of shape so that they could help induce iterative improvement of hypothesis during the solving process. We compare and present the results for our proposed GenSearch algorithm and the baselines against the same 38 images.

For the human experiments, the MNIST task included 10 participants, and 11 people participated in the Fashion MNIST task, with no overlap between the two groups. We later asked 6 people each from the same group to solve the respective tasks at a very high difficulty level.

4.2.1. Comparison of categorical classification

We estimated the performance of the GenSearch to solve the constellation task by measuring the classification accuracy of its output (see Fig 9b). We see that the algorithm classified images with an accuracy of 0.66 (+-0.04) on MNIST and 0.63 (+-0.03) on Fashion MNIST. In comparison, humans performed relatively close with an accuracy of 0.59(+-0.08) on MNIST and 0.61 (+-0.06) on Fashion MNIST. As the classes in Fashion MNIST dataset are imbalanced, we calculated balanced accuracy for this set which is 0.64 for GenSearch and 0.59 for humans. We observe that the model confuses similar things as humans, for instance mistaking objects with similar features, like Pullover and Shirt for a T-shirt or Dress for a Coat. In MNIST, numbers such as 2 and 8 are often confused. In general confusion matrix is

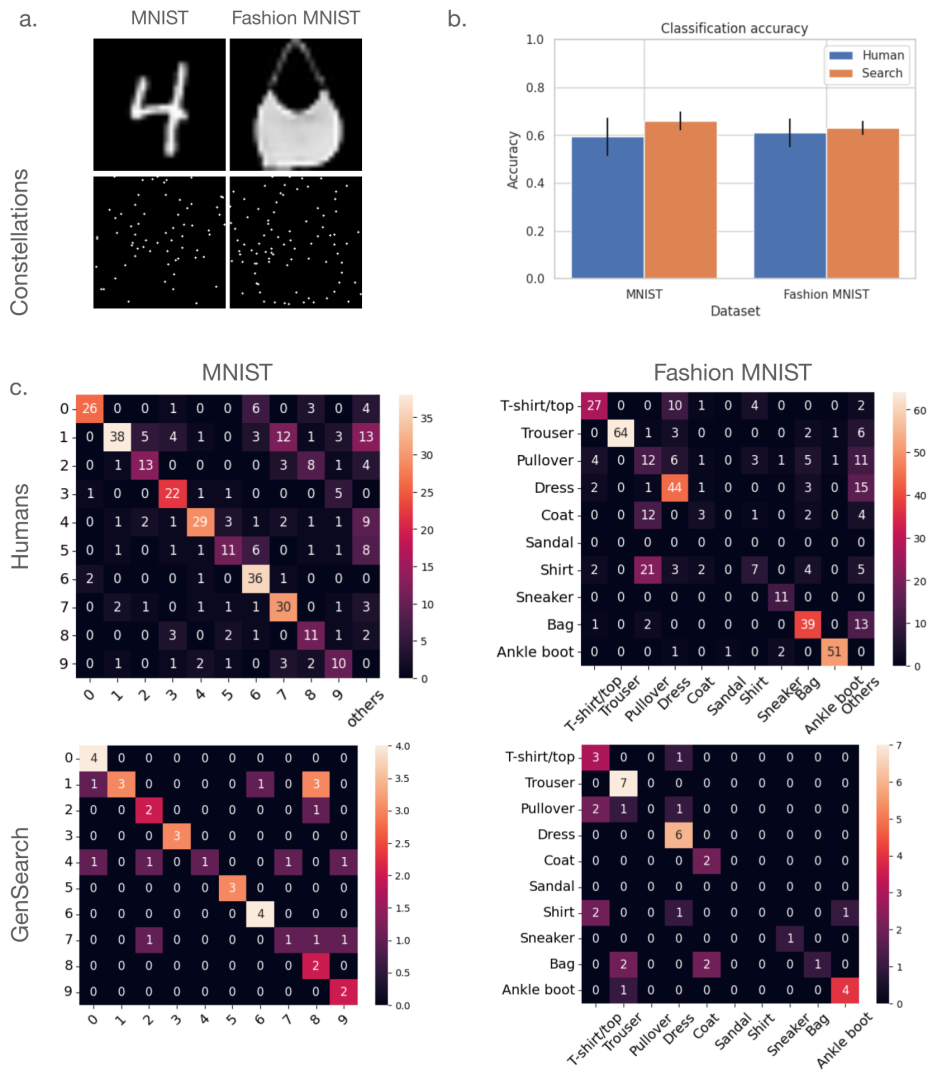


Figure 9: Classifying objects in constellation images: a) Sample images from the original Fashion MNIST and MNIST datasets, along with their constellations version below them. b) The plot shows the comparable classification accuracy of humans and GenSearch algorithm on a test set of 38 images on both datasets c). Confusion matrix for classification in 1) Fashion MNIST for humans 2) Fashion MNIST for GenSearch. The correlation between the two matrices is 0.79. Confusion matrix for 3) MNIST for humans and 4) MNIST for search algorithm with a correlation of 0.70 between the matrices. We remove the entries when the corresponding elements are 0 in both matrices to calculate the correlation. Additionally, in many cases, humans don't respond or give a response that does not correspond to any of the given options. Such responses are aggregated in the column 'Others'.

more different between humans and GenSearch as most numbers differ from each other by only 1 edge. The correlation between the confusion matrix (in Fig 9c) for humans and GenSearch is high (excluding common zeros), with 0.79 for Fashion MNIST and 0.70 for MNIST. The total counts of human confusion matrixes have all the participants responses counted directly hence the total responses are 38 $\times$ number of participants.

4.2.2. Comparison of drawings made by GenSearch and Humans

In the process of finding the correct shape hidden in the constellation images, we prompted humans to draw their solution outlines. They could erase and redraw some solutions if they found them unsatisfactory. GenSearch also generated and matched its solutions to the constellations during the solving process. Hence, we can obtain these drawn solutions and try to compare them. To quantify the alignment of drawings produced by humans and GenSearch, we introduced a metric called IOU(Dots), which is similar to the IOU metric used by the image segmentation community to quantify the alignment of the segmented region to the true region. In IOU dots, we identified the dots passed by the drawing solutions of humans and GenSearch, and then quantified the overlap in those dots using the formula given below:

$$IOU(dots) = \frac{n(A \cap B)}{n(A) + n(B) - n(A \cap B)}$$

where the number of dots that the contour in drawing A passes through (with a margin of 3 pixels for all solutions) is n(A), and the number of dots passing through solution B is n(B). The number of dots in both solutions is $n(A \cap B)$. IoU(dots) was chosen over metrics like IOU over drawings because it prioritizes the overlap calculation over shared evidence (the dots) rather than penalizing the stylistic variations of the hand-drawn lines.

In Fig 10, we illustrate the process of calculation of IOU(dots) over two constellation images and their respective solutions for the MNIST and Fashion MNIST dataset. We note that over the set of 38 test images, the overlap between the solutions by humans and GenSearch is high with IOU(dots) 0.49 for Fashion MNIST and 0.6 for the MNIST dataset. In comparison, the baseline IOU(dots) between ground truth outline and human drawings is 0.69 for Fashion MNIST and 0.76 for MNIST.

These numbers indicate a higher human–model agreement on digits than on complex clothing items. This difference is likely attributable to dataset difficulty and structural complexity: digits possess more rigid topologies that restrict the space of plausible hypotheses. In contrast, the Fashion-MNIST objects (e.g., "pullover" or "dress") have more fluid boundaries and higher intraclass variation in dots arrangement, leading to a wider diversity of partial hypotheses between

Figure 10: Examples from Fashion-MNIST (top) and MNIST (bottom) of left to right 1) Constellation image; 2) dots covered by human drawing; 3) dots covered by search algorithm drawing; 4) common dots between search and human drawing. The intersection over union (dots) for this Fashion MNIST is 0.25, whereas IOU(dots) for MNIST is 0.8. The overall IOU(dots) for the whole dataset for human drawings to the corresponding search solution is 0.49 for Fashion MNIST and 0.6 for the MNIST dataset.

humans and GenSearch.

4.2.3. Evolution of solutions

In Fig 11 we observe the process followed by GenSearch to solve the constellation images as iterative refinement over the search generations. The top solution for each generation of the population is shown in the figure over the iterations. We see how the top solution is iteratively optimised to the structure of the dots, trying to find shapes that pass through the maximum number of dots. The nature of the evolution of the top solution is similar to how humans try to optimise their candidate solutions while solving the constellation images. In the initial stages, major changes are made to the shape to try to find the best shape category that suits the structure of the dots. As seen in the top panel of Fig 11, sometimes humans start with the correct solution but then change the solution to a slight variation. We see that the GenSearch solution can also change between the same categories. Overall, we see a similar pattern of refinement in solutions between humans and GenSearch. The only difference remains where humans can sometimes identify the solutions easily at first glance, the initial solution pool of GenSearch remains random; hence, matching human performance at this stage depends on randomly having the correct or a near correct shape available in the initialised population of 1000 candidates. Overall, by visualizing these intermediate states, we have

evidence that GenSearch can mimic the human behavioral pattern of starting with a coarse global shape and gradually adjusting local features to match sensory evidence. This iterative refinement process aligns with the emerging view that generative inference provides a unified framework for feedback processing in both natural and artificial vision [97].

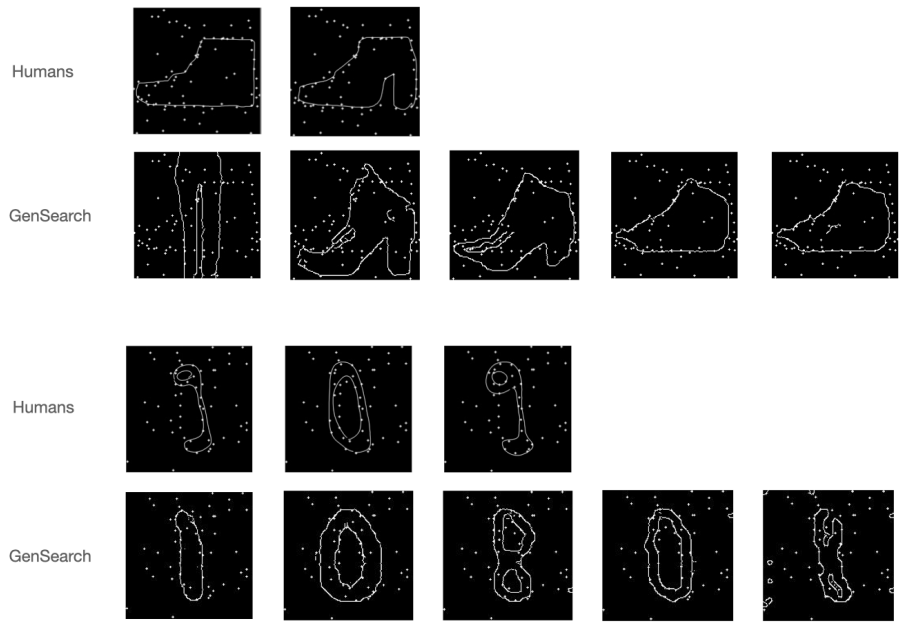


Figure 11: Qualitatively, the process of solving for humans and GenSearch looks similar as both processes iteratively change their top solution in the search process. The figure depicts an instance where both humans and GenSearch explore similar candidate solutions. The correct solution for the top image is a shoe and for the bottom image is the number 1.

4.2.4. Maintaining multiple hypotheses

The Gensearch algorithm uses evolutionary search as its search mechanism, which, by design, works by having a pool of multiple candidate hypotheses. In the previous section, we saw how sometimes the top solution for the GenSearch algorithm has the tendency to immediately change from an unrelated category to the correct one. In this task, human participants also often report the sudden emergence of a top solution—a "eureka" moment where a previously ambiguous cluster of dots suddenly crystallizes into a recognizable object. In Fig 12, we show an instance of such a solving attempt by GenSearch, but also show, during the process, 5 of the top 200 candidate solutions selected by the evolutionary search at various generations. We see how the top solution remains relatively stable from the first generation to the 15th generation, with minor adjustments in shape to better fit

the dots, but the correct solution, i.e., 2, appears in the 25th generation. We can see that some prototype of the candidate solution 2 always existed in the solution pool from generation 1 and is getting strongly represented in the top 200 solution pool in the later generations. The shape of 2 gets refined while participating in this pool, while finally matching the shape in the constellation image and emerging as the top solution. These dynamics of maintaining and refining multiple solutions in the process can be a natural candidate to explain phenomenon we observe in humans, such as the sudden emergence of a top solution.

Furthermore, the maintenance of diverse candidates in the latent space relates to the phenomenon of multi-stable solutions in perception, such as the Necker Cube or the Rubin Vase. In these classic studies, the human visual system flips between two or more equally plausible interpretations of the same sensory input. By maintaining a population of latent vectors, GenSearch effectively models this multi-stability; the algorithm does not commit to a single interpretation until the evidence (the dot overlap) strongly favors one structural hypothesis over the others.

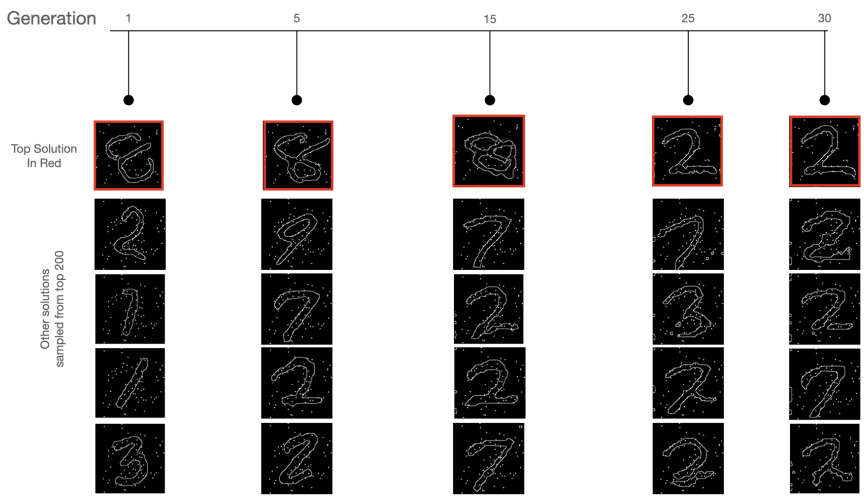


Figure 12: In this example, we show how the correct solution (number 2) appears in the candidate pool of GenSearch before becoming the top solution. The correct solution appears already in generation 1 and with increased frequency in generations 5 and 15 before converging as the top solution in generation 25.

4.3. Comparison of GenSearch variations (Evolutionary vs Gradient-based Search)

We formulated a version of GenSearch using gradient descent as the search algorithm instead of evolutionary search. As the objective used by evolutionary search is to maximise the number of dots passed through the outline, we formulated

a differentiable version of loss that closely resembled the original objective function. For this, we created a heatmap around the constellation dots with the same pixel tolerance as the one used to count the dots passing near an edge. The heatmap has values 1 for the dots and 0 for non-dots regions. As we use a GAN as the image generator, it can already calculate gradients through the network. We made the edge detection process by using convolution of the GAN-generated images with Sobel filters. The GAN output processed by Sobel filters gives us the edge map.

The dot-product similarity loss between a target dot heatmap $D(x,y)$ and the generated edge map $E(x,y)$ is defined as:

$$\mathscr{L} = -\sum_{x,y} D(x,y) \cdot E(x,y)$$

This optimisation is equivalent to increasing the number of edges passing through the dots. Finally, we optimised this loss using gradient descent using a triangular learning rate schedule.

We observe in Table 1 that the solutions by the gradient descent search algorithm are crossing a higher number of dots in the constellation image but the accuracy of the search is fairly worse. Here, the accuracy is defined as the categorical classification accuracy of the final optimised solutions. We observe in the solutions of the search algorithm that the GenSearch algorithm over-fits to spurious shapes and solutions that maximise the number of dots. This also suggests a regularising effect of the probabilistic sampling and heuristic-based search and composition in evolutionary search using mutations and crossovers.

Table 1: Accuracy and average dots covered for difficulty level 11 using gradient descent to find the optimal solution

	MNIST		**Fashion MNIST**	
Search Variant	**Accuracy**	**dots covered**	**Accuracy**	**dots covered**
Genetic Search	0.66	33	0.63	54
Gradient Descent	0.18	49	0.23	75

4.4. Comparison with Other Machine Learning Models

We compare the results of GenSearch with other machine learning models such as CLIP, Resnet 18, and Pix2Pix. All these models represent a different class of model, where CLIP represents a pre-trained vision language model that can be used for zero-shot classification on images. Resnet18 is a CNN model that can be trained and evaluated on the image classification task. Pix2pix is an image generation model that we trained to translate between the constellation image and the ground-truth object outlines.

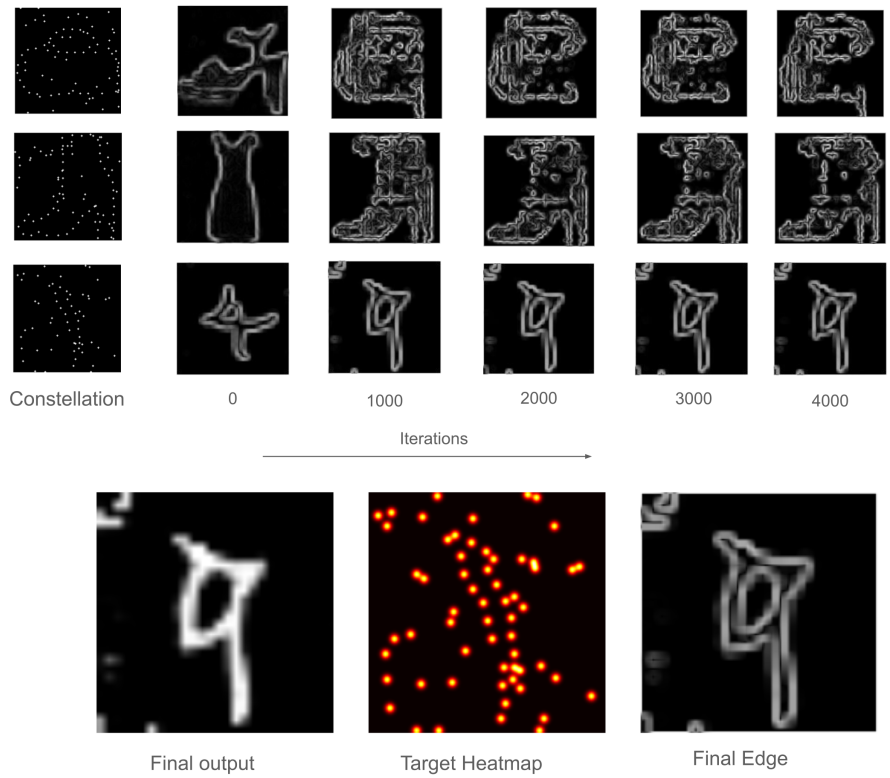


Figure 13: Solving process of gradient-based search: The upper panels show for different constellation images the change in the candidate updated using gradient descent over the iterations. The lower panel shows the 'Final output' i.e. the final constellation image generated by the search. 'Target heatmap' i.e. the heat map of dots that the edges of the output should try to pass through and the 'Final edge' showing the edge map corresponding to the 'Final output'.

4.4.1. Zero-shot classification with CLIP

CLIP is a pre-trained model trained by aligning the representation of an image to its corresponding text description. As this model can perform zero-shot classification of images, we use the model to classify the constellation images in the MNIST and Fashion MNIST constellation test sets. CLIP classifies these images with a near-random accuracy of 0.07 and 0.11 for MNIST and Fashion MNIST, respectively. The performance for corresponding original MNIST and Fashion MNIST sets is 0.42 and 0.68 respectively. The low performance on constellations set is likely due to CLIP never being trained on images in form of constellations.

4.4.2. Trained Resnet18 models

We evaluated Resnet18 models that were trained on 60,000 training images

and were evaluated for the 38 test images used in the comparison with GenSearch and Humans. Overall, we see in Fig 14a that these models achieve a near perfect accuracy above 0.9 which degraded with difficulty level but still remained far above the human performance on these images. This sufficientlhy higher accuracy on even higher difficulty levels may point to these models learning a different algorithm as compared to the ones employed by humans and GenSearch.

We attributed this superhuman performance to these models' over-fitting to the training data. To explicitly test this, we evaluated the models trained on a difficulty level on all other difficulty levels. We can see in Fig 14b and Fig 14c, the explicit effect of overfitting as the model performance degrades when evaluated on difficulty levels other than the training level. Further, we see this surprising effect that the model with very high accuracy on the very difficult level 17 performs fairly poorly on the easy difficulty level 9, indicating the over-fitting to a specific difficulty level by these trained models.

4.4.3. Pix2pix: Image-to-image translation models

Pix2pix is an image-to-image translation model which can learn to generate an image in modality B conditioned on a corresponding image in modality A. In our case, we use this model to translate between the constellation image and the corresponding object outlines. These models were trained on the training set of the MNIST and Fashion MNIST constellation datasets and evaluated on the 38 test set images. We can see from the results in Table 2 using the IOU (dots) metric that the outlines generated by these models did not match the human drawings as well as the GenSearch's solutions. Further, the mistakes made by these models also did not overlap well with the mistakes made by humans, as seen by the IOU (dots) mistakes metric. In an additional analysis, using the model trained on the MNIST dataset on the Fashion MNIST test set revealed that overlap (IOU dots: 0.23) is similar to the dataset specific models, indicating that the current performance does not rely on learning the model of the object shapes in the dataset but could be learning some local heuristic rules to connect dots at certain distance.

Table 2: Average IOU(dots) and IOU(mistakes) between machine and human drawings. IOU(dots) between human drawings and ground truth is 0.69 and 0.76 for MNIST and Fashion MNIST, respectively.

	MNIST		**Fashion MNIST**	
Model	**IOU (dots)**	**IOU (mistakes)**	**IOU(dots)**	**IOU(mistakes)**
GenSearch	0.6	0.43	0.49	0.58
Pix2Pix	0.23	0.21	0.25	0.32

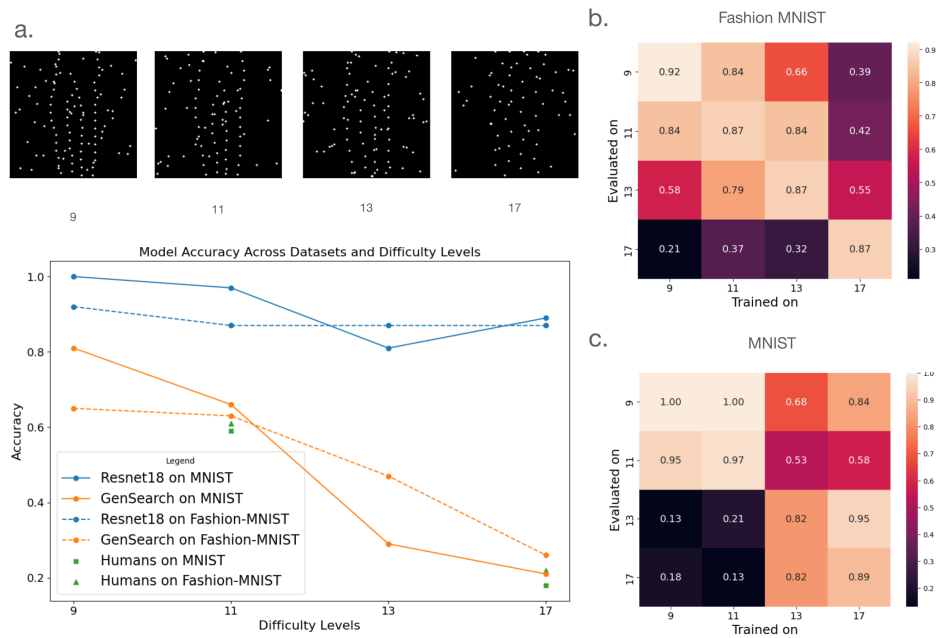


Figure 14: a) The difficulty level of the constellation images is controlled by changing the distance between the dots, as shown in the example images in the figure. The plot shows the performance of the models and humans on different difficulty levels. Observe how the Resnet18 train on a particular difficulty level has much better classification accuracy compared to GenSearch and humans, especially on datasets with higher difficulty, i.e., level 17. b) ResNet 18 performance on change of train/test distribution for Fashion MNIST. Where the x-axix denotes the difficulty level the model is trained on and y-axis denotes the difficulty level of the evaluation set. The heatmap values represent the accuracy for a particular training-evaluation difficulty setting. Note how performance is unstable even on lower difficulty levels as it deviates from the training distribution. c) ResNet 18 performance on change of train/test distribution for MNIST

4.5. Discussion and main takeaway

In this chapter, we compared the performance of a generative search algorithm (GenSearch) to humans on the complex visual problem-solving task of finding objects hidden in constellation images. We observed that humans and GenSearch algorithm show similarity in terms of overall accuracy, types of mistakes, drawing overlaps and overlap in mistakes.

In our results, the performance disparity between ResNet18 and GenSearch highlights a fundamental trade-off between discriminative efficiency and generative robustness. While ResNet18 achieves higher peak accuracy on fixed difficulty settings, this success may stem from the model learning task-specific shortcuts

by using local pixel correlations within the training distribution. In contrast, GenSearch offers value through its human-aligned failure modes and interpretability. This distinction is further emphasized by the near-random performance of CLIP, which confirms that the constellation modality effectively strips away the local textures and semantic features that modern Vision-Language models rely on.

A critical insight gained from comparing search strategies is the objective mismatch which can be observed in the gradient-based variant of GenSearch. Although the gradient-based approach often achieves higher dot coverage and lower reconstruction loss, it gives significantly worse classification accuracy than the evolutionary variant. This behavior indicates that the gradient optimizer overfits to the objective by distorting the latent representation to capture outlier noise dots, eventually exiting the manifold of GANs. Conversely, the population-based diversity of the evolutionary algorithm acts as a form of regularization, preventing the model from committing to unnecessarily complex outlines. This reveals that the "sudden emergence" of a top solution is more robustly managed through a diverse hypothesis space.

Expanding on the robustness of the evolutionary approach, it is essential to consider the nature of the walk performed through the generator's embedding space. Because the latent manifold of a GAN is continuous, any optimization trajectory, whether gradient-based or evolutionary traverses an interpolated states between candidate classes. For instance, a search path might move through a morph solution that is visually halfway between a Pullover and a Coat, possessing ambiguous features of both. While the gradient-based variant often settles into these unrecognizable, over-fitted hybrids to maximize dot coverage, the evolutionary algorithm's reliance on a population of discrete latent vectors may better approximate the categorical nature of human perception. Human participants rarely report seeing stable hybrid objects; instead, they experience bistable flips between discrete, familiar hypotheses. By maintaining a diverse pool of candidates rather than a single moving point, GenSearch mimics this bistability, allowing the system to maintain multiple competing pure categories. This suggests that the evolutionary mechanism does not just regularize the complexity of the outline, but also preserves the semantic integrity of the solution by avoiding the low-probability regions between established object manifolds.

The maintenance of multiple simultaneous hypotheses within an evolutionary framework has some parallels to the architecture of the biological brain. In a connectionist system, the population of candidates used by GenSearch can be mapped onto distinct neural ensembles or population codes that compete and cooperate to explain sensory input [98]. The concept of crossover finds a biological analog in the integration of partial solutions across different cortical columns [99]. Overall the evolutionary mechanism used in GenSearch aligns with theories of Neural Darwinism, where neural groups undergo a selection like process based on the fit between their activity and the incoming sensory data [100]. In terms of behaviour, maintaining a library of candidate solutions that can improve while

interacting with each other is a way to explain such sudden shifts where the correct solution is found suddenly instead of iteratively converging to a solution class, which closely resembles human 'aha moments' with sudden insights [101, 102].

Overall, the findings should promote further research into GenSearch-like implementations of analysis-by-synthesis focusing on aspects of evolutionary search which shows benefits like probabilistic regularisation and explanation of phenomena like multi-stable vision and insights. This further offers a general framework that allows fits between high-level concepts space and low-level image structure.

5. INTERPRETING THE REPRESENTATIONS OF DL BASED VISION ENCODERS ON MULTI-OBJECT NATURAL SCENES

As we have seen, feed-forward machine vision models such as CLIP, ResNet18, and pix2pix exhibit differences in their solving behaviour compared to humans in terms of accuracy and performance under distribution shifts. While analysis on a synthetic dataset such as constellations provides useful insights about robustness and utilisation of part-whole hierarchy, it is relevant to understand the performance of more ubiquitously used DL models on natural scenes. The scene, in terms of the whole-part hierarchy, can be thought of as a combination of various objects. We have observed in the previous sections how the combination of parts is crucial in describing an object and how the search to find the object is dependent on the combination of parts. To observe if similar computations combining different objects into a coherent scene gist take place in a vision encoder as it encodes natural scenes, we analysed the representation and processing of multi-object scenes in popular vision encoders.

While many representation learning methods for vision encoders report classification accuracy as a measure of representation ability. These evaluations are still biased towards single-label or single-object classification on datasets such as ImageNet. Evaluation using multi-object scenes and performance metrics will show the interations between the different objects and how they are represented in the network. VLMs trained with objectives such as image-text matching or caption generation have reported difficulties in properly representing relationships between scene objects [103, 54]. While these studies report poor performance on explicit modelling of relationships between the objects, we evaluate individual object's representations in terms of its decodability in the network according to their relationship with the scene and the other objects.

To evaluate the vision encoders for this property, we created a paired object decoding task by creating sets of images from the COCO [86] dataset that contain pairs of certain objects in a scene. The task allows us to check the representation of these objects in relation to each other throughout the encoder network. We further use the object segmentation mask from the COCO dataset to identify object specific tokens in the network and use the human-annotated image captions to identify objects that the humans would find important in describing the images.

Our analysis of several popular vision encoders trained with different objective functions revealed that, when evaluated on paired-object tasks, these network's output tends to represent main objects of the scene better at the cost of secondary objects in the background. While this result is natural in accordance to the training objectives of these networks which requires them to focus on certain objects given a scene, it raises questions about the general-purpose usability of these vision encoders in models where they are expected to perform equally well on a variety

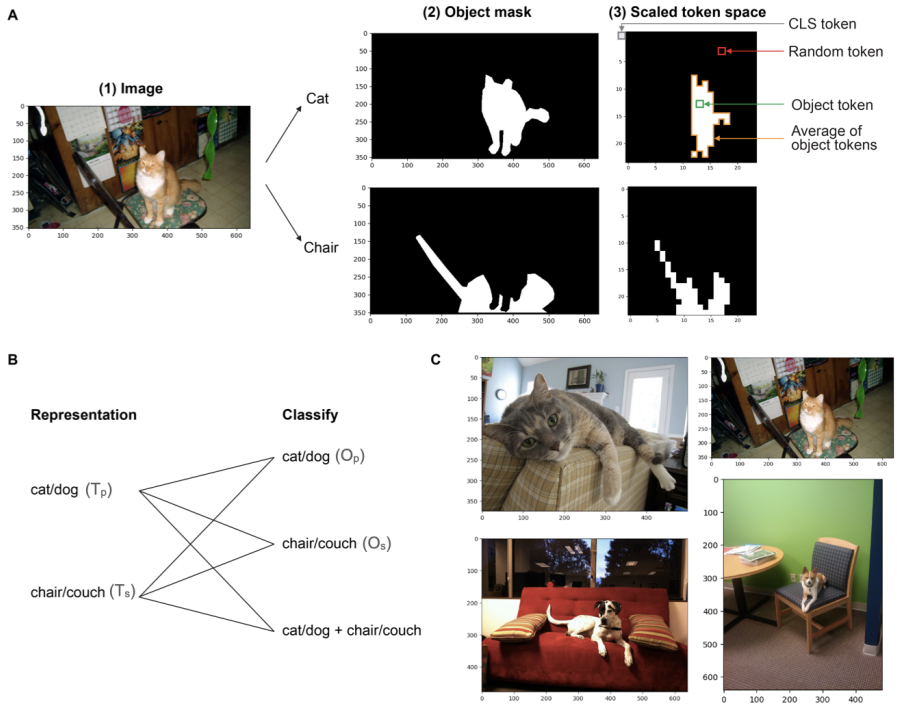


Figure 15: **A.** Explanation of how the token representations are obtained. We analyse four kinds of tokens in this study: 1) CLS token: The special token usually used in models for downstream tasks; 2) Avg_obj(Object-specific token): obtained by averaging the token representations of the object-masked tokens, as shown in the figure. 3) Random_obj (Object-specific token): Rather than averaging, we sample one of the tokens from the masked token space of the object 4) Random: Obtained by sampling any random token from the token space other than the CLS token **B.** Describes the experimental setup in which we perform decoding in paired object tasks; each object-specific representation decodes 1) the object itself, 2) the other object in the image, and 3) the combination of both objects. **C.** Shows a sample paired object decoding task; given an image, the task is to decode if it contains object1 (cat/dog), object2 (chair/couch) or a combination of both.

of objects in the scene such as in VQA and robotics. In this chapter, we will discuss the details of this setup and how it can lead to some general insights regarding the working of vision encoders and their adaptation for downstream tasks. The code and the data splits used in this chapter are available at : `https://github.com/tarunkhajuria42/Structured-representations`

5.1. Proposed decoding tasks and the used dataset

To gain insight into how object-wise information is represented in various layers and tokens of a pre-trained transformer, we used a probing setup. A probe is a model (usually small and simpler) which quantifies the decodability of a property or set of classes from an internal representation of the model being analysed. The probing setup in this study quantifies a linear model 's ability (measured by hold out set accuracy) to learn to distinguish a type of object from others by using the representation at a layer/token. The linearity of the model ensures that the features corresponding to particular set of objects being decoded were already linearly separable. So measuring the probe's performance to classify certain scene objects at various layers and tokens of a vision encoder, helps us understand how those objects are represented at those layers and tokens. An aggregate view of such performance helps us visualise the variation in object representation across the vision encoder's different layer and at different kinds of tokens.

The primary evaluation is a paired-object decoding task, designed to test how the encoder represents multiple entities simultaneously. For each image, we define two tiers of objects:

1. Primary Objects: The most salient entities likely to dominate a human description (e.g., Cat vs. Dog).
2. Secondary Objects: Contextual entities that co-occur with the primary subjects but are often less central to the scene (e.g., Bench, Chair, Couch, or Bed).

By using a linear model, we can ask three specific questions of the representation:

1. Does this token identify the Primary Object (e.g., is it a cat or a dog)?
2. Does this token identify the Secondary Object (e.g., is there a chair or a bed)?
3. Does the representation capture the Combination of both (e.g., a "dog" on a "couch")?

We selected six distinct sets of objects from COCO dataset (detailed in Table 3), where primary and secondary objects have high co-occurrence. For instance, in Set 1, we test the model's ability to distinguish a cat from a dog while simultaneously identifying whether the scene contains a chair or a sofa. This forces the probe to determine if the encoder has successfully bound a certain amount of object specific properties together, helping to distinguish them from other similar objects. We also sub-sample to balance the co-occurrence of all object combinations between the primary and secondary categories. This balancing step is important to measure entanglement correctly, as otherwise, the difference in the co-occurrence of primary and secondary objects allows the decoder to learn the presence of one from the other.

We analyze four specific locations within the network to determine where object data is most accessible (also shown in Fig 15a):

Table 3: For paired object decoding, we use 6 object sets with different numbers of images in each set. Each set contains images with different variations of objects. For the control global object decoding task, we tested generalisation on 20 randomly chosen objects.

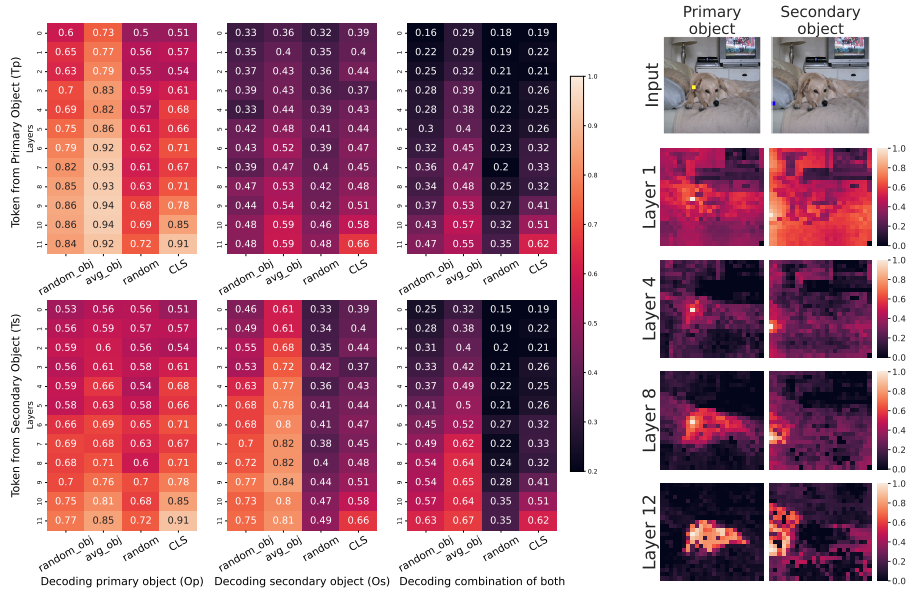
Paired-Object Probe	**Set 1**	**Set 2**	**Set 3**	**Set 4**	**Set 5**	**Set 6**
# Images	2414	5042	1953	2143	938	3738
Primary Object	cat, dog	dining table, person	train, bus	tv, laptop	microwave, oven	motorcycle, car
Secondary Object	bench, chair, couch, bed	pizza, knife, cup, cake	traffic light, bench, backpack, handbag	mouse, remote, keyboard, cellphone	bottle, spoon, knife, cup	traffic light, handbag, backpack, bicycle
Global Probe	sheep, bear, banana, potted plant, bowl, toilet, horse, apple, fire, parking meter, handbag, snowboard, broccoli, giraffe, stop sign,hydrant, cow, tie, hot dog, truck, wine glass					

1. CLS Token: The global token typically used for downstream classification.
2. Avg_Obj: The average vector of all tokens located within the object's segmentation mask.
3. Random_Obj: A single token sampled from the object's mask to test if local features are sufficient.
4. Random: A baseline token sampled from the background (areas not belonging to the primary or secondary objects).

To evaluate if the observed representation patterns extend beyond the specific paired-object scenarios, we implemented a global decoding task. The methodology utilizes 20 randomly sampled objects from the COCO dataset, distinct from those used in the paired-object experiments. The global decoding task involves training linear probes to identify the presence of these 20 categories using the object specific representations i.e. Avg_Obj.

5.2. Representation of objects in the layers and token space of networks

Using the decoding task with image sets described in the previous section, we obtained figures describing the average decoding accuracy across the six object sets. An example of such a figure is given in Fig 16.a for the model BLIP. Observing



(a) (b)

Figure 16: **a.** Paired object decoding task results for BLIP across layers: Average decoding performance for different layers (y-axis) and token types (x-axis) over 6 tasks for BLIP. In the subfigures, the y-axis contains variations of where the object-specific tokens (random_obj and avg_obj) are obtained. The different columns show results for 1) decoding the primary object; 2) the secondary object and 3) the combination of both objects in the image. The decoding pattern remains after averaging, with the tokens from the objects modelling the most useful information for categorising the objects. The object-specific tokens are much better than the CLS token, which has to capture the larger scene context. **b.** Visualisation of cosine similarity of highlighted token to other tokens for a token from primary and secondary objects at various layers of BLIP model.

this figure, one can make some important inferences about the structure of object-specific information distributed across the network. We observed that object-specific decoding accuracy increases down the layers until a certain layer for all token types in the network. The highest decoding accuracy for the tokens was usually given by the object-specific tokens in the network rather than the CLS token. The CLS token encoded a more entangled scene level representation with high accuracy for both primary and secondary objects, hence its object-specific representation was lower than the object-specific tokens. On the other hand, we observed that these object-specific token representations were themselves not completely token-wise disentangled. It was possible to decode the other object in the scene from the object-specific token representations (Avg_Obj and Random_Obj tokens).

In an instance level view of the representations, we observed how the models were representing a single image at various layers in Fig 16.b. Here, in the first image we took a token from the object 'dog' and in the second we took a token from the object 'bed'. We then calculated and plotted the cosine similarity of those tokens with all other tokens of the network. We saw that as the representations are processed by the network in the later layers, the token originating from these objects becomes similar to the selected token representation. In the later layers, they seem to almost form a segmentation mask for the object in the image. This information, in conjunction with the improvement of object-specific classification performance in the later layers of the network, indicates that the tokens bind information down the layers. We also see which parts are considered as one at the last layer of the network, as the 'bed' token originating from the 'pillow' shows equal similarity to the 'bed' in the intermediate layers but more specific similarity to the 'pillow' in the last layer.

5.3. The pattern of representation across networks: Key Insights

While comparing the pattern of representations across many models, we employed a linear probing setup to decode object categories from specific token representations. In Fig 17, we see that the relative accuracy trends for various tokens were similar across most Vision-Language Models (VLMs). A notable observation was the higher overall decoding performance of FLAVA, BLIP, and BLIP-Large models, which also showed a higher segregation of information; specifically, the classification accuracy of a random background token is significantly lower in the last layer compared to the object-specific tokens, indicating that these models successfully localize object features. The two self-supervised models, DINO and DINOv2, showed a slightly lower accuracy for secondary objects, yet their representations were more disentangled, which allowed them to model the background objects relatively better. This disentanglement is evidenced by DINOv2 achieving a higher decoding accuracy for randomly sampled object tokens (approximately 90% for primary objects) than most other models, aligning with its documented utility in semantic segmentation tasks.

We observed a difference in the structure of representations due to architectural constraints (Transformer vs. CNN) and specific training on multi-object tasks. Consequently, there was a significant decrease in decoding accuracy using a random CNN unit representation compared to random ViT tokens; specifically, for primary objects, the ViT achieved 0.84 accuracy versus 0.72 for the CNN, while for secondary objects, the ViT reached 0.54 versus 0.45 for the CNN. Further, the object-specific tokens in CNNs had lower accuracy while decoding the other objects in the scene than their ViT counterparts. This suggests that CNNs exhibit less entanglement of information across objects because they lack a global attention mechanism, which forces object features to remain physically localized within

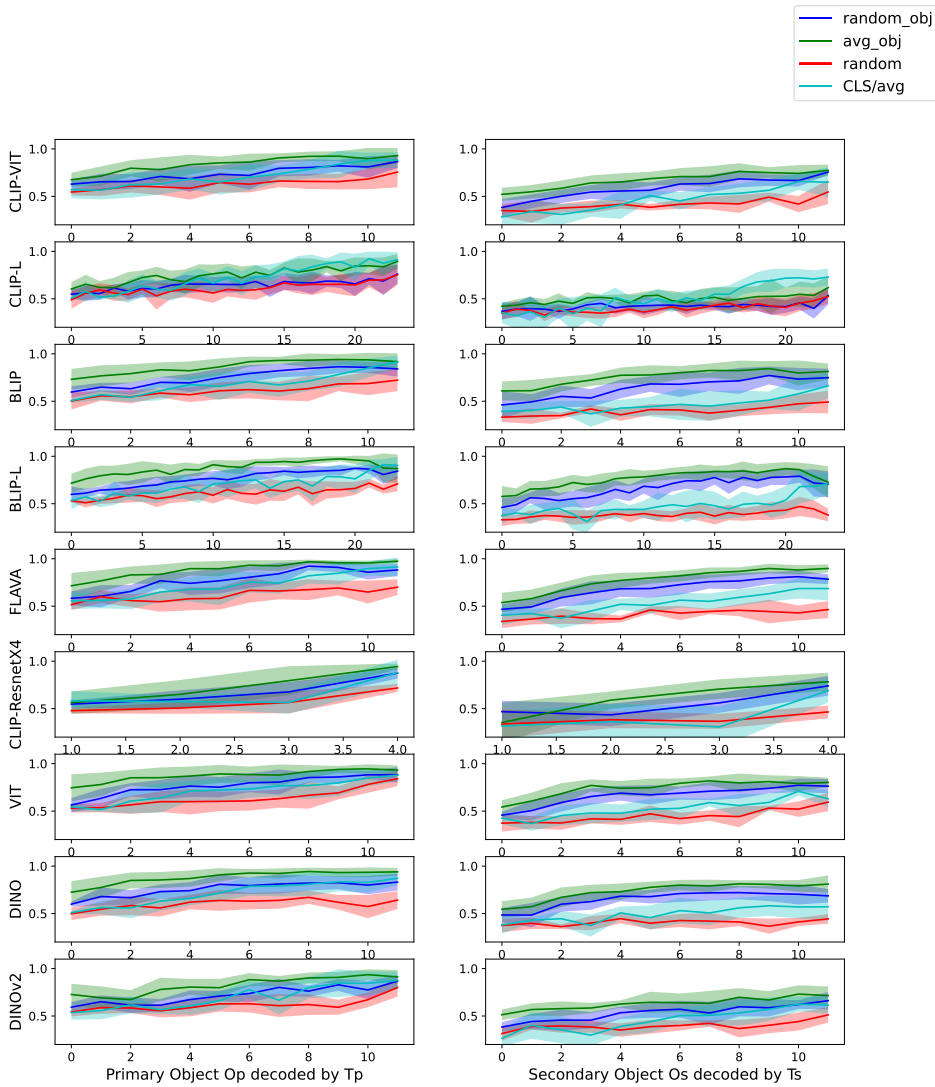


Figure 17: Layer-wise test set decoding accuracy for primary and secondary objects for pre-trained models in the study. Where O_p,O_s denotes primary and secondary object category, while T_p, T_s denotes token from primary and secondary objects. The accuracies are averaged over the six object sets. In each sub-graph, the y-axis denotes the decoding accuracy, and the x-axis denotes the layer at which the accuracy was observed. We observe consistent decoding trends across models with a few variations reported in Section 5.3.

specific units.

In contrast, the ViT trained on ImageNet-21k showed the least differentiation between object-specific tokens and background tokens. In this model, a random token decodes the primary object with an accuracy of 0.84, which is nearly identical

to the 0.88 accuracy of the global CLS token. This indicates that in models trained only for single object classification, scene-level information appears more uniformly dispersed across all tokens because the network was never required to distinguish between multiple co-occurring objects. This result highlights that the segregation of information in the last layer is more explicit in VLMs because their training objectives such as image-text matching or captioning, require the correct modeling of multiple distinct objects.

Lastly, the decoding accuracy for object-specific tokens degraded the most on secondary objects, which was also correlated with lower accuracy on objects not mentioned in the human-annotated captions. Our layer-wise analysis shows that in BLIP-Large and several other networks, including ViT, DINOv2, and CLIP, the accuracy for secondary objects begins to decrease in the final layers while primary object accuracy remains stable. This "late-layer fade" suggests that the training objective forces the network to compress the scene representation into a simplified narrative, effectively suppressing secondary details to focus on the primary subjects of the scene.

5.4. Difference in object representation based on object importance

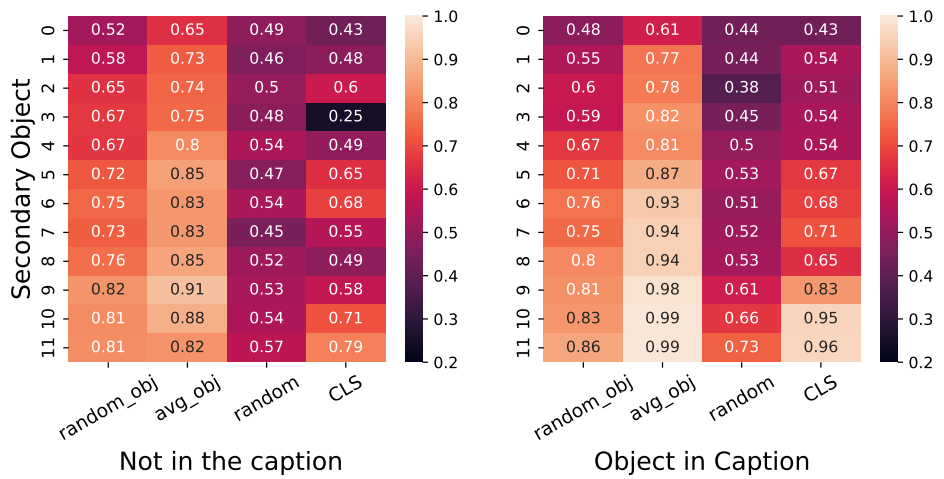


Figure 18: Variation in decoding accuracy between instances of objects ‘in caption’ and ‘not in caption’. Each subplot represents the decoding of the object by its object-specific representation. The heatmap represents the decoding accuracy for a particular token type at a particular layer. Where the x-axis represents the various token types and the y-axis represents the layers from which the particular representation is obtained. The two heatmaps compare the two conditions in which the representations of objects are divided based on if they are ’not in caption’ or if they are mentioned ’in caption’

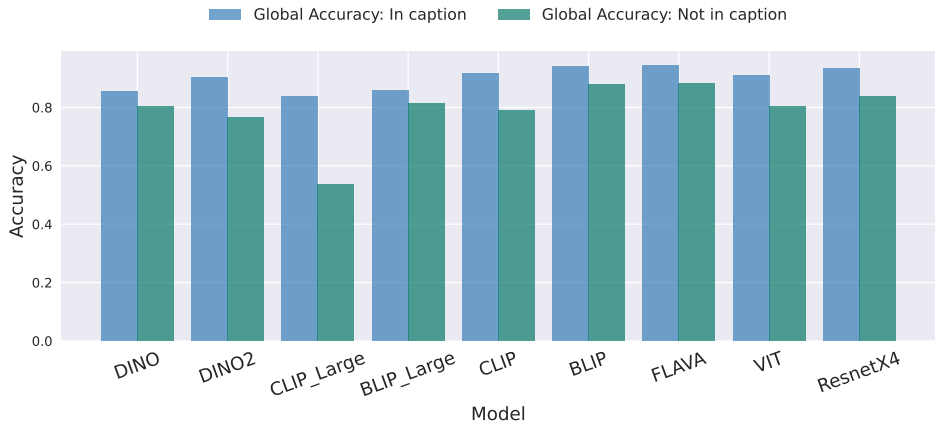


Figure 19: The figure shows the final decoding accuracy on the 20-class global decoding task for objects when they are mentioned 'in caption' as compared to when they are 'not mentioned in caption'. We observe that across all networks at the last layers objects mentioned 'in caption' can be decoded linearly as compared to other objects. Certain networks like CLIP Large show this effect to a greater degree but they have better object-specific representation in the CLS tokens.

Finally, we evaluated the difference in representation in the network of objects included 'in caption' compared to objects 'not mentioned in the caption' in the COCO dataset. Our assumption was that the objects that are not mentioned in the human descriptions of the scene are not as important for the scene. We observe the difference in their representation in two views. In Fig 18 we see for BLIP how the representation throughout the network has fairly less decoding accuracy than the similar positions in the network when the objects are mentioned in the caption. In particular, note how the CLS token, which is directly optimised by the objective function, is the most affected in this comparison final decoding accuracy of CLS is 0.79 when object is 'not in caption' vs 0.96 when objects are 'in caption'. Finally, in Fig 19 we see that this effect is general across all networks when we replicated it using a 20-class global probe across all networks in this study.

5.5. Discussion and main takeaways

In this chapter, we learned how the object-specific information binds to individual tokens as it is processed down the layers of the vision encoders. In a multi-object scene, the CLS token tends to abstract away information about some objects and represent a few main objects in the scene better than others. Other objects have better representations in the object-specific tokens in the network. This leads us to the main takeaway from our analysis that not all objects in the scene are equally represented in the final output layers of the network. For many objects, some of the earlier layers have a better decodable representation of those objects. This understanding has implications for optimal use of these object encoders, as in

various models intended for general downstream tasks, such as LVLMs for general question answering or VLAs for generating action-related commands. For example, models such as LLaVA[104] report better overall performance using earlier layers of the pre-trained CLIP vision encoder.

Beyond these technical implications, these results provide crucial insights regarding robust scene understanding. If the final layer of a vision encoder acts as a bottleneck that prioritizes primary objects while suppressing secondary ones, a single-pass feed-forward approach is fundamentally insufficient for complex, real world tasks. This finding supports the broader claim presented in this thesis that human-like robustness requires an ability to re-examine early layer features when the task demands interaction with other objects.

Furthermore, the importance bias identified in this chapter suggests that the current gap between machine and human vision is not just a matter of classification accuracy, but a difference in efficient information abstraction and composition. While humans can flexibly shift focus across the whole-part hierarchy, modern encoders are locked to a specific viewpoint as dictated by their training captions. This chapter therefore complements the previous chapters, where we explore how iterative mechanisms and feedback loops can compensate for this inherent filtering, ensuring that the vision system remains robust even when the objects of interest are not the primary focus of the initial scene description.

6. DISCUSSION

One of the main goals of modern deep learning research is to create models that perform their intended task robustly. While complete robustness to variation in inputs or tasks is not possible, maximum coverage of the possible scenarios and robustness to critical failures is the more pragmatic goal. As many deep learning applications are centered around the tasks performed by humans, aligning with humans becomes a goal for the field to maximise usability and safety in scenarios that humans encounter.

Cognitive scientists have long tried to build models to explain human cognition and behaviour. While such models captured aspects of neural measures or behavioural data, they remained brittle, hence not being able to explain a wide variety of data or even capture human behaviour very well [105]. Deep Learning models, when optimised to perform certain tasks, are one of the few models that perform close to human performance [37, 106]. Further, the alignment of their internal representations to neural data in many modalities has sparked interest in using the understanding of these models to refine the questions we ask in cognitive science [107].

In this thesis, we have explored this synergy between cognitive science and deep learning to study scene understanding. With the constellation task, we explicitly tried to highlight the iterative aspect implicit in vision. While human vision in everyday settings constantly works under the influence of current context (environment, tasks or goals), many of the computer vision models are just processing the image once in a feed-forward manner. In this thesis, we explore why this might not be the best strategy.

To further elaborate the difference between computer vision models and human vision during scene perception, we gained evidence from human experiments on the constellation task, demonstrating how humans can iteratively search and generate the solution even on a fairly underspecified image such as a constellation. This task, in its essence, shows the robustness of our vision as we are not trained on these images but can still infer the hidden shape in an input with very low signal about the object. On the other hand, our analysis of most feed-forward networks such as CLIP, pix2pix or ResNet18 suggests that these networks cannot adapt to this change of modality without explicit training on the task. Further analysis of networks such as ResNet18 that work supernaturally better than humans on this task suggests that these networks overfit to a particular level of sparsity in dots that they are trained on and cannot adapt to changes in these sparsity levels, even towards a fairly easier level for the same constellation set.

The generative search algorithm (GenSearch) performed close to humans under the constraints of no explicit training on the modality, specifically when searching in a limited semantic space. The interplay between search and the bottom-up fit evaluation allowed for search to be continued for a dynamic period until there was no improvement over a few iterations. This mechanism supports more robust

transfer from natural images to the constellation images, displaying the natural pattern of better performance on easier images. Another important dynamic that the iterative search captures is that it enables the integration of cues at various levels of abstraction, similar to human vision. For example, some iteration could optimise for a fit for curves but another would also find a higher level fit with object parts and the complete object shape. This shift allows for the search to be guided by fits at various levels of shape abstraction in the part-whole hierarchy, making use of as many clues as possible to solve the image. While this search process does not solve the images as efficiently as humans, the implicit use of iterative inference during search shows the role of effectively collecting and refining information at various scales for more robust visual inference.

In contrast to this flexible search, when we evaluate the feed-forward vision encoders trained for various objectives on natural images that have multiple objects, we observe that they rigidly focus on certain objects in the image more than others. While this property is desirable for the objectives for which these models are originally trained, they may not serve as a good general-purpose scene encoder, as they may not equally represent the details for all objects in the output embedding or token space.

In this interplay between the scene structure and its contents, the importance of correctly inferring elements at various scales of the part-whole hierarchy becomes crucial. This ability is more important in very ambiguous inputs like constellations, where it may be difficult to infer the whole object at once, hence one has to rely on collecting cues in the form of object parts.

In a multi-object scene, objects become parts of the scene, although they are typically more loosely coupled compared to the parts of an object. The deep learning based image encoders treat certain objects in the scene differently than others, modelling some relationship between the objects in terms of their importance. However, as discussed above, it can be suboptimal to use these models in general-purpose downstream tasks, due to their redefined scene-object importance, which may not match the object importance structure for every downstream task. The encoders learn to represent various objects at different levels of abstraction in their output representation. Although for many models the information is not completely lost and the downstream network can still recover the correct representation for its purpose which, however, may cost additional training data.

The findings from the representational analysis of multi-object scenes provides a critical appreciation for the generative search approach (GenSearch). Where the analysis in Chapter 5 about vision encoders revealed that feed-forward encoders often exhibit a salience bias, where representations for smaller or background objects are suppressed. GenSearch using a multi-hypothesis search, its generative component and relying on low level matching (using counting) allows a more thorough sampling of inputs to provide a more robust scene explanation. By applying these insights from GenSearch, we can hypothesize that human vision resolves the representational crowding found in machine encoders by using the iterative cycle

to suppress irrelevant features and generate missing details of obscured objects. This suggests that human representations are not static embeddings like those in CLIP or DINO, but are instead dynamic, generative constructs that are constantly updated. This understanding of dynamic contextual embedding also hints at reasons for the relative success of autoregressive LLMs and VLMs over static models. Future research could investigate whether the sudden emergence of a solution in the constellation task corresponds to a representational shift in the brain.

6.1. Limitations, opportunities and implications for understanding human and computer vision

Some of the main contributions of this work are identifying and highlighting computational components that may better replicate human behaviour in solving vision tasks. Through our work on constellations, we highlight the benefits of an evolutionary search strategy to maintain and refine hypotheses under the analysis-by-synthesis framework. Some of the key benefits we noted are its natural regularising effect which prefers more likely shapes while ignoring noisy artefacts. Further, this search strategy, which maintains and refines multiple hypothesis solutions in its library, provides a natural explanation for the sudden emergence of correct solutions during many human solving attempts. While we are not suggesting that the brain runs an evolutionary search algorithm in this form, certain computational elements from the search, such as the maintenance of a library of solutions, some stochastic operations for composition and conditional updating based on crossover and mutation, are worth exploring in future research. On the other hand, future research directions need to address our main limitation of large computing costs and inference times, which makes the algorithm inefficient to scale for more natural settings, such as the Things constellations [22]. Some promising future directions in this case are the use of more hybrid approaches, including faster feed-forward guidance [14] and the inclusion of recurrent or gradient-based updates to form hybrid search strategies.

While we have already considered some of the popular deep learning models as baselines, there are a few likely candidates that can be explored in the future work. A recurrent descriminate model can be a useful baseline in addition to current feedforward baseline with Resnet18. Similarly, our initial experiments adapting Sketch RNN [108] to constellations task did not give promising results hence was not explored further, however this remains a promising research direction as sketch models and specially a recursive generative model shall be an efficient alternative to traditional analysis-by-synthesis models. In terms of VLMs, we evaluated CLIP which encodes images in a feed-forward manner, while the modern VLMs such as GPT5 or LLaVA [104] have a generative and iterative next word prediction loop, giving them the ability to look back at the image information multiple times during the process. Evaluation of such models on constellations datasets in the future may provide useful insights about robustness in scene understanding.

Our work analysing vision encoders is also limited in terms of the inferences that can be made based on linear probing tasks. In particular, the lower accuracy for certain objects does not necessarily imply that information about the object is not present in those tokens. Similarly, a higher accuracy on a few class categorisation tasks may not imply complete binding of category-specific information in a particular token. Nevertheless, the decoding trends across the layers and in tokens are still informative. Moreover, the main results regarding better object-specific representations have been tested on a downstream task [23] and are consistent with reports in a few other studies [104, 109]. However, future work can further test the actionability of the layer and token-specific inferences for various encoders by using them accordingly in downstream tasks or as a frozen vision encoder for a VLM.

In general, we find through this work that inspecting representations for the correct level of abstraction can be quite informative about a model's behaviour. While we focused on object-specific abstractions in this study, the application of these principles to other forms of scene abstractions can also reveal useful insights. In the human visual system, the ventral stream learns object-specific invariances of parts, leading to the understanding of the category of specific objects or scene elements. While in the dorsal stream, the representations abstract away the exact object identities but maintain various scene-level information about objects, their arrangements, and their importance to the task, as well as information regarding action-related affordances of the objects. This perspective suggests that being able to extract information about various aspects of the scene in a segregated manner, so it is accessible to processes involved in reasoning about the scene, can be useful for generalisation. However, the vision encoders tested in our analysis revealed that the models represent category-related information, but the position of the objects and their relative size and arrangement are implicitly represented in terms of its distribution of category information in the tokens of the network. We speculate that this lack of explicit segregation of category and position information contributes to the limitation of many current VLMs, which while performing well on questions about scene contents still struggle with tasks that require content independent algorithms, such as counting and searching [54]. In the current models, we see how different models trained with a particular objective and architecture tend to prioritise a particular type of information. VLMs trained in conjunction with language prompts are better at maintaining high-level semantic information about the scenes and their contents, but can struggle with spatial relationships in the scene [104, 55]. The DINO family of models trained to learn an invariant scene representation under various transformations, learn both the semantic relationships along with some spatial properties of the scene, such as depth [56]. The third family of models such as SAM which are trained to learn an even more local objective of segmenting individual parts of the scenes, are known to have low semantic understanding of the objects they can segment [110]. Although these model families are improving their deficiencies with each update, the fundamental

question still remains whether the intended general purpose deep learning models can actually form and represent information in the abstract form for many tasks, which would further indicate them implicitly using learning and using content independent algorithms.

Overall, the work in this thesis attempts to evaluate the computational components that underlie the robustness of human vision in difficult conditions. Further, through the identification and adaption of such computational components and the general view of analysing models in terms of its representations, this thesis hopefully contributes to the development of more robust machine vision models.

7. CONCLUSION

In this thesis, we explored the topic of scene understanding in human and machine vision as a process that involves more than just a single feed-forward pass, but rather an active iterative process of collecting information about a scene at various levels of detail. We introduced a new dataset inspired by star constellations and showed that it can induce iterative inference in human participants solving this task. We attempted to replicate the human strategies used to solve this task by developing a generative search algorithm. This allowed us to identify computational components such as multi-hypothesis maintenance and refining in evolutionary search, which could help better replicate aspects of human behaviour in computational models. The viewpoint of understanding multi-component representation helped us to probe vision encoders used in DL models, which led to identifying problems with the optimal use of the static encoder outputs for general downstream tasks. Overall, by finding specific computational components that support the general principles of iterative inference and the use of part-whole hierarchy, this thesis adds to our knowledge of better ways to model human scene understanding. Additionally, these computational findings and the view of analysing representations in the form of object-specific abstractions also informs better engineering and analysis of machine vision models.

BIBLIOGRAPHY

[1] Moshe Bar. "Visual objects in context". In: *Nature Reviews Neuroscience* 5.8 (2004), pp. 617–629.

[2] Aude Oliva and Antonio Torralba. "The role of context in object recognition". In: *Trends in cognitive sciences* 11.12 (2007), pp. 520–527.

[3] Daniel J Felleman and David C Van Essen. "Distributed hierarchical processing in the primate cerebral cortex." In: *Cerebral cortex (New York, NY: 1991)* 1.1 (1991), pp. 1–47.

[4] Courtney J Spoerer et al. "Recurrent neural networks can explain flexible trading of speed and accuracy in biological vision". In: *PLoS computational biology* 16.10 (2020), e1008215.

[5] Daniel Kaiser and Radoslaw M Cichy. "Parts and wholes in scene processing". In: *Journal of Cognitive Neuroscience* 34.1 (2021), pp. 4–15.

[6] Aude Oliva. "Building the gist of a scene: The role of global image features in recognition". In: *Visual Perception, Part 2* (2006), pp. 23–36.

[7] David Navon. "Forest before trees: The precedence of global features in visual perception". In: *Cognitive Psychology* 9.3 (1977), pp. 353–383.

[8] Courtney J Spoerer, Patrick McClure, and Nikolaus Kriegeskorte. "Recurrent convolutional neural networks: a better model of biological object recognition". In: *Frontiers in psychology* 8 (2017), p. 1551.

[9] Drew Linsley et al. "Learning long-range spatial dependencies with horizontal gated recurrent units". In: *Advances in neural information processing systems* 31 (2018).

[10] Tim C Kietzmann et al. "Recurrence is required to capture the representational dynamics of the human visual system". In: *Proceedings of the National Academy of Sciences* 116.43 (2019), pp. 21854–21863.

[11] Kohitij Kar et al. "Evidence that recurrent circuits are critical to the ventral stream's execution of core object recognition behavior". In: *Nature neuroscience* 22.6 (2019), pp. 974–983.

[12] Alan Yuille and Daniel Kersten. "Vision as Bayesian inference: analysis by synthesis?" In: *Trends in cognitive sciences* 10.7 (2006), pp. 301–308.

[13] Brenden M Lake et al. "Building machines that learn and think like people". In: *Behavioral and brain sciences* 40 (2017), e253.

[14] Ilker Yildirim et al. "Efficient inverse graphics in biological face processing". In: *Science advances* 6.10 (2020), eaax5979.

[15] Hakan Yilmaz et al. "Seeing in the dark: Testing deep neural network and analysis-by-synthesis accounts of 3D shape perception with highly degraded images". In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. Vol. 43. 43. 2021.

[16] Benjamin Recht et al. "Do cifar-10 classifiers generalize to cifar-10?" In: *arXiv preprint arXiv:1806.00451* (2018).

[17] Dan Hendrycks and Kevin Gimpel. “A baseline for detecting misclassified and out-of-distribution examples in neural networks”. In: *arXiv preprint arXiv:1610.02136* (2016).

[18] Robert Geirhos et al. “Generalisation in humans and deep neural networks”. In: *Advances in neural information processing systems* 31 (2018).

[19] Robert Geirhos et al. “Shortcut learning in deep neural networks”. In: *Nature Machine Intelligence* 2.11 (2020), pp. 665–673.

[20] Sara Beery, Grant Van Horn, and Pietro Perona. “Recognition in terra incognita”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 456–473.

[21] Tarun Khajuria et al. “Constellations: A novel dataset for studying iterative inference in humans and AI”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 5142–5152.

[22] Tarun Khajuria, Kadi Tulver, and Jaan Aru. “Comparing a computational model of visual problem solving with human vision on a difficult vision task”. In: *PLOS Computational Biology* 21.12 (2025), e1012968.

[23] Tarun Khajuria et al. “Interpreting the structure of multi-object representations in vision encoders”. In: *World Conference on Explainable Artificial Intelligence*. Springer. 2025, pp. 359–382.

[24] John M Findlay and Iain D Gilchrist. *Active vision: The psychology of looking and seeing*. 37. Oxford University Press, 2003.

[25] David Milner and Mel Goodale. *The visual brain in action*. Vol. 27. Oup Oxford, 2006.

[26] Kurt Koffka. *Principles of Gestalt Psychology*. 1st. New York: Harcourt, Brace and Company, 1935. URL: https://doi.org/10.4324/9781315009292.

[27] Anne M Treisman and Garry Gelade. “A feature-integration theory of attention”. In: *Cognitive psychology* 12.1 (1980), pp. 97–136.

[28] Jeremy M Wolfe, Kyle R Cave, and Susan L Franzel. “Guided search: an alternative to the feature integration model for visual search.” In: *Journal of Experimental Psychology: Human perception and performance* 15.3 (1989), p. 419.

[29] Jeremy M Wolfe. “Guided Search 6.0: An updated model of visual search”. In: *Psychonomic bulletin & review* 28.4 (2021), pp. 1060–1092.

[30] David H Hubel and Torsten N Wiesel. “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex”. In: *The Journal of physiology* 160.1 (1962), p. 106.

[31] Ichiro Fujita et al. “Columns for visual features of objects in monkey inferotemporal cortex”. In: *Nature* 360.6402 (1992), pp. 343–346.

[32] Melvyn A Goodale and A David Milner. “Separate visual pathways for perception and action”. In: *Trends in neurosciences* 15.1 (1992), pp. 20–25.

[33] Christopher Summerfield, Fabrice Luyckx, and Hannah Sheahan. “Structure learning and the posterior parietal cortex”. In: *Progress in neurobiology* 184 (2020), p. 101717.

[34] Dwight J Kravitz et al. “A new neural framework for visuospatial processing”. In: *Nature Reviews Neuroscience* 12.4 (2011), pp. 217–230.

[35] Vladislav Ayzenberg and Marlene Behrmann. “The dorsal visual pathway represents object-centered spatial relations for object recognition”. In: *Journal of Neuroscience* 42.23 (2022), pp. 4693–4710.

[36] Kunihiko Fukushima. “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”. In: *Biological cybernetics* 36.4 (1980), pp. 193–202.

[37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012).

[38] Kaiming He et al. “Mask r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.

[39] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. “Densecap: Fully convolutional localization networks for dense captioning”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 4565–4574.

[40] Jason Yosinski et al. “How transferable are features in deep neural networks?” In: *Advances in neural information processing systems* 27 (2014).

[41] Yoshua Bengio, Aaron Courville, and Pascal Vincent. “Representation learning: A review and new perspectives. arXiv 2012”. In: *arXiv preprint arXiv:1206.5538* (2012).

[42] Kaiming He et al. “Momentum contrast for unsupervised visual representation learning”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 9729–9738.

[43] Ting Chen et al. “A simple framework for contrastive learning of visual representations”. In: *International conference on machine learning*. PmLR. 2020, pp. 1597–1607.

[44] Alec Radford et al. “Learning transferable visual models from natural language supervision”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8748–8763.

[45] Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).

[46] Angela Radulescu, Yeon Soon Shin, and Yael Niv. “Human representation learning”. In: *Annual Review of Neuroscience* 44.1 (2021), pp. 253–273.

[47] Junnan Li et al. “Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 12888–12900.

[48] Mathilde Caron et al. “Emerging properties in self-supervised vision transformers”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 9650–9660.

[49] Maxime Oquab et al. “Dinov2: Learning robust visual features without supervision”. In: *arXiv preprint arXiv:2304.07193* (2023).

[50] Alexander Kirillov et al. “Segment anything”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2023, pp. 4015–4026.

[51] Aphex34. *Typical CNN architecture*. File: Typical cnn.png. Wikimedia Commons. Original date: 16 December 2015. Licensed under CC BY-SA 4.0. 2015. URL: https://commons.wikimedia.org/wiki/File:Typical_cnn.png (visited on 12/15/2025).

[52] Daniel Voigt Godoy. *Vision Transformer*. File: Vision Transformer.png. Wikimedia Commons. Original date: 6 June 2021. Source: https://github.com/dvgodoy/dl-visuals/. Licensed under CC BY 4.0. 2021. URL: https://commons.wikimedia.org/wiki/File:Vision_Transformer.png (visited on 12/15/2025).

[53] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[54] Declan Campbell et al. “Understanding the limits of vision language models through the lens of the binding problem”. In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 113436–113460.

[55] Shiqi Chen et al. “Why is spatial reasoning hard for vlms? an attention mechanism perspective on focus areas”. In: *arXiv preprint arXiv:2503.01773* (2025).

[56] Shir Amir et al. “Deep vit features as dense visual descriptors”. In: *arXiv preprint arXiv:2112.05814* 2.3 (2021), p. 4.

[57] Diederik P Kingma and Max Welling. “Auto-encoding variational bayes”. In: *arXiv preprint arXiv:1312.6114* (2013).

[58] Ian J Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems* 27 (2014).

[59] Aston Zhang et al. *Generative adversarial network*. File: Generative adversarial network.svg. Wikimedia Commons. Original date: 7 December 2023. Source: https://github.com/d2l-ai/d2l-en. Licensed under CC BY-SA 4.0. 2023. URL: http://commons.wikimedia.org/wiki/File:Generative_adversarial_network.svg (visited on 12/15/2025).

[60] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising diffusion probabilistic models”. In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851.

[61] Robin Rombach et al. “High-resolution image synthesis with latent diffusion models”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 10684–10695.

[62] Stanislaw Antol et al. “Vqa: Visual question answering”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2425–2433.

[63] Ranjay Krishna et al. “Visual genome: Connecting language and vision using crowdsourced dense image annotations”. In: *International journal of computer vision* 123 (2017), pp. 32–73.

[64] Justin Johnson et al. “Clevr: A diagnostic dataset for compositional language and elementary visual reasoning”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2901–2910.

[65] Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. “On the binding problem in artificial neural networks”. In: *arXiv preprint 2012.05208* (2020).

[66] Ellie Pavlick. “Symbols and grounding in large language models”. In: *Philosophcal Transactins of the Royal Society* 381.2251 (2023), p. 20220041.

[67] Charles Lovering and Ellie Pavlick. “Unit testing for concepts in neural networks”. In: *Transactions of the Association for Computational Linguistics* 10 (2022), pp. 1193–1208.

[68] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. “Neural discrete representation learning”. In: *arXiv preprint arXiv:1711.00937* (2017).

[69] Francesco Locatello et al. “Object-centric learning with slot attention”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 11525–11538.

[70] Akshay Aravindan et al. “Do VLMs Have Bad Eyes? Diagnosing Compositional Failures via Mechanistic Interpretability”. In: *ICCV Workshop on Explainable Vision-Language Models*. 2025.

[71] Zhiqiu Lin et al. “Revisiting the Role of Language Priors in Vision-Language Models”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2025.

[72] Shuo Yang et al. “Look-back: Implicit visual re-focusing in mllm reasoning”. In: *arXiv preprint arXiv:2507.03019* (2025).

[73] Danfei Xu et al. “Scene graph generation by iterative message passing”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 5410–5419.

[74] Shimon Ullman et al. “Human-like scene interpretation by a guided counter-stream processing”. In: *Proceedings of the National Academy of Sciences* 120.40 (2023), e2211179120.

[75] Baifeng Shi, Trevor Darrell, and Xin Wang. “Top-down visual attention from analysis by synthesis”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 2102–2112.

[76] Daniel LK Yamins et al. “Performance-optimized hierarchical models predict neural responses in higher visual cortex”. In: *Proceedings of the national academy of sciences* 111.23 (2014), pp. 8619–8624.

[77] Ilya Kuzovkin et al. “Activations of deep convolutional neural networks are aligned with gamma band activity of human visual cortex”. In: *Communications biology* 1.1 (2018), p. 107.

[78] J Redmon et al. “You only look once: Unified, real-time object detection. arXiv 2015”. In: *arXiv preprint arXiv:1506.02640* (2015).

[79] Netta Ollikka et al. “A comparison between humans and AI at recognizing objects in unusual poses”. In: *arXiv preprint arXiv:2402.03973* (2024).

[80] Minkyu Choi et al. “A dual-stream neural network explains the functional segregation of dorsal and ventral visual pathways in human brains”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 50408–50428.

[81] Vonne Van Polanen and Marco Davare. “Interactions between dorsal and ventral streams for controlling skilled grasp”. In: *Neuropsychologia* 79 (2015), pp. 186–191.

[82] Jessica AF Thompson et al. “Zero-shot counting with a dual-stream neural network model”. In: *Neuron* 112.24 (2024), pp. 4147–4158.

[83] Guillaume Alain and Yoshua Bengio. “Understanding intermediate layers using linear classifier probes”. In: *arXiv preprint arXiv:1610.01644* (2016).

[84] John Hewitt and Percy Liang. “Designing and interpreting probes with control tasks”. In: *arXiv preprint arXiv:1909.03368* (2019).

[85] Ruben S van Bergen and Nikolaus Kriegeskorte. “Going in circles is the way forward: the role of recurrence in visual inference”. In: *Current Opinion in Neurobiology* 65 (2020), pp. 176–193.

[86] Tsung-Yi Lin et al. “Microsoft coco: Common objects in context”. In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[87] John Canny. “A computational approach to edge detection”. In: *IEEE Transactions on pattern analysis and machine intelligence* 6 (1986), pp. 679–698.

[88] Martin N Hebart et al. “THINGS: A database of 1,854 object concepts and more than 26,000 naturalistic object images”. In: *PloS one* 14.10 (2019), e0223792.

[89] Martin N Hebart. “The THINGS initiative: a global initiative of researchers for representative sampling of objects in brains, behavior, and computational models”. In: *Journal of Vision* 22.14 (2022), pp. 3203–3203.

[90] Mathias Eitz, James Hays, and Marc Alexa. “How Do Humans Sketch Objects?” In: *ACM Trans. Graph. (Proc. SIGGRAPH)* 31.4 (2012), 44:1–44:10.

[91] Kevin Frans, LB Soros, and Olaf Witkowski. "Clipdraw: Exploring text-to-drawing synthesis through language-image encoders". In: *arXiv preprint arXiv:2106.14843* (2021).

[92] Peter Schaldenbrand, Zhixuan Liu, and Jean Oh. "StyleCLIPDraw: Coupling Content and Style in Text-to-Drawing Translation". In: *arXiv preprint arXiv:2202.12362* (2022).

[93] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[94] Han Xiao, Kashif Rasul, and Roland Vollgraf. "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms". In: *arXiv preprint arXiv:1708.07747* (2017).

[95] David J Field, Anthony Hayes, and Robert F Hess. "Contour integration by the human visual system: evidence for a local "association field"". In: *Vision research* 33.2 (1993), pp. 173–193.

[96] Priyank Jaini, Kevin Clark, and Robert Geirhos. "Intriguing properties of generative classifiers". In: *arXiv preprint arXiv:2309.16779* (2023).

[97] Tahereh Toosi and Kenneth D Miller. "Generative inference unifies feedback processing for learning and perception in natural and artificial vision". In: *bioRxiv* (2025), pp. 2025–10.

[98] Alexandre Pouget, Peter Dayan, and Richard Zemel. "Information processing with population codes". In: *Nature Reviews Neuroscience* 1.2 (2000), pp. 125–132.

[99] György Buzsáki. *Rhythms of the Brain*. Oxford University Press, 2006.

[100] Gerald M Edelman. *Neural Darwinism: The theory of neuronal group selection*. Basic Books, 1987.

[101] Kadi Tulver et al. "Restructuring insight: An integrative review of insight in problem-solving, meditation, psychotherapy, delusions and psychedelics". In: *Consciousness and cognition* 110 (2023), p. 103494.

[102] Kadi Tulver, Karl Kristjan Kaup, and Jaan Aru. "The road to Aha: A recipe for mental breakthroughs". In: *Cognition* 257 (2025), p. 106081.

[103] Mert Yuksekgonul et al. "When and Why Vision-Language Models Behave like Bags-Of-Words, and What to Do About It?" In: *The Eleventh International Conference on Learning Representations*. 2022.

[104] Haotian Liu et al. "Visual instruction tuning". In: *Advances in neural information processing systems* 36 (2023), pp. 34892–34916.

[105] Danko Nikolić. *The Crisis of Computational Neuroscience*. Sapien Labs. 2019. URL: https://sapienlabs.org/the-crisis-of-computational-neuroscience/.

[106] David Silver et al. "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529.7587 (2016), pp. 484–489. DOI: 10.1038/nature16961.

[107] Adrien Doerig et al. “The neuroconnectionist research programme”. In: *Nature Reviews Neuroscience* 24.7 (2023), pp. 431–450.

[108] David Ha and Douglas Eck. “A neural representation of sketch drawings”. In: *arXiv preprint arXiv:1704.03477* (2017).

[109] Daniel Bolya et al. “Perception encoder: The best visual embeddings are not at the output of the network”. In: *arXiv preprint arXiv:2504.13181* (2025).

[110] Miguel Espinosa et al. “There is no SAMantics! Exploring SAM as a Backbone for Visual Understanding Tasks”. In: *arXiv preprint arXiv:2411.15288* (2024). arXiv: `2411.15288 [cs.CV]`.

ACKNOWLEDGEMENTS

I would like to sincerely thank my family, supervisor, mentors, reviewers, colleagues, friends and many other people who have helped me in direct and indirect ways during my work and life. I acknowledge the use of LLM during the editing process of this thesis.

SISUKOKKUVÕTE

Stseeni mõistmine inim- ja arvutinägemises

Inimese nägemine on äärmiselt kohanev ja vastupidav muutuvatele oludele. Me suudame oma nägemist tõhusalt kasutada erinevatel eesmärkidel autojuhtimisest kuni lugemise ja sulgpalli mängimiseni, ning mitmekesistes oludes, näiteks tuule, tolmu, osalise varjatuse ja valguse muutumise korral. Kognitiivteaduses kirjeldatakse seda nägemise generatiivse olemuse kaudu: silmadest tulnud alt-üles-signaalid ja aju kõrgema kognitsiooni piirkonnad mõjutavad üksteist vastastikku. Nende kahte tüüpi signaalide korduv vastastikune mõju võimaldab meil tuvastada ja ajakohastada konteksti ning seejärel tõlgendada visuaalseid sisendeid selle konteksti alusel. See protsess toetabki inimese nägemise vastupidavust ja kohanemisvõimet.

Paljud tänapäeva arvutinägemise süsteemid kasutavad aga ainult pärilevi (ingl feedforward) tehisnärvivõrke. Kuigi need mudelid suudavad kiiresti ja üsna täpselt täita spetsiifilisi treenitud ülesandeid, ei ole nad sobivad, et saavutada head tulemust treeningjaotusest väljaspool olevate sisendite ja ülesannete puhul. Uuringud on välja toonud, et need mudelid tuginevad ülesannete täitmisel liialt madala taseme tunnustele ja tekstuuridele ning on altid õppima otseteid näiliste korrelatsioonide kujul. Selle tulemuseks on vähem vastupidav nägemissüsteem, mis ei laiene treeningjaotusest erinevatele sisenditele.

Selles doktoritöös uurisime inimese ja masinnägemise sarnasusi ja erinevusi stseeni esitamisel ning mõistmisel.

Esimeses peatükis kavandasime tähtkujudest inspireeritud keerulise nägemisülesande, kus inimesed ja masinalgoritmid püüdsid leida pildil peidetud objekti, millel on vähe signaale objekti määratlemiseks. Ülesanne loodi optimaalse raskusastmega, kus inimestel on keeruline objekti täpimustrites kiirelt edasisuunatud viisil leida, võimaldades meil seega uurida iteratiivset järeldusahelat, mis hõlmab ülevalt-alla ja alt-üles visuaalsete signaalide vastastikmõju. Meie katsetes märkisid paljud inimestest osalejad ülesande lahendamise ajal mitme hüpoteesi moodustamist ja täpsustamist. Katsed eeltreenitud pildikeelemudeliga CLIP näitasid ligikaudu juhuslikku sooritusvõimet, kuna neid pole spetsiaalselt sellisteks visuaalseteks sisenditeks treenitud. Kokkuvõttes avaldasime paljudest andmekogumitest, nagu THINGS, Sketch, MNIST ja Fashion MNIST, tähtkuju-stiimulite versioonid, mida aktiivselt kasutatakse kognitiivteaduse ja arvutinägemise uurimisel. Lisaks avaldasime ka oma koodi, mida teadlased saavad kasutada oma piltide teisendamiseks tähtkuju versioonideks oma uurimisvajaduste järgi.

Teises peatükis panime kokku generatiivse otsingualgoritmi (nimega GenSearch), et lahendada tähtkuju pilte, mis sisaldavad numbreid ning objekte MNIST ja Fashion MNIST andmekogumitest. See algoritm teeb otsingu nendel andmekogumitel treenitud pildigeneraatori (GAN) latentses ruumis ja püüab leida pildi, mille kontuur sobib tähtkuju punktidega kõige paremini. Seejärel võrdlesime seda algoritmi inimestega, kes

lahendasid samu pilte, ning leidsime palju huvitavaid sarnasusi, kuidas algoritm teeb vigu, säilitab mitut hüpoteesi ja ajab sarnaseid objekte segamini. Nii inimesed kui ka GenSearch lahendasid need ülesanded umbes 60% täpsusega. Testisime ka konvolutsioonilisi tehisnärvivõrke (CNN-e), treenides neid tähtkujude piltidel, ja märkasime CNN-ide ebaloomulikku sooritust. Nägime tähtkujude piltide eri raskusastmete ja sama algoritmi seadistuse põhjal, et inimesed ja GenSearch varieeruvad sarnaste mustritega. Vastupidi sellele tuli CNN-e treenida vastavalt piltide raskusastmele, mis viitab mudelite ülesobitamisele (ingl overfitting). Selle peatüki peamised järeldused puudutavad evolutsiooniliste algoritmide kasutamist otsingul. Me leidsime, et geneetilise otsingu tõenäosuslik olemus, mis säilitab samal ajal lahenduste kogumit, näitas regulariseerimisefekti eeliseid ahne gradientlaskumise otsingu ees. See selgitab loomulikult ka mõne inimesest osaleja õige lahenduse ootamatu esilekerkimise nähtust.

Viimaks testisime me praegu süvaõppes kasutusel olevaid nägemiskoodreid COCO andmestiku loomulike, mitut objekti sisaldavate stseenidega. Disainisime kahe objekti esitamise ja sondeerimise ülesande, et analüüsida koodrite kihtides ja märgitüüpides (ingl token type) leiduvaid esitusi. Me tuvastasime märgid, mis on seotud objekti asukohaga, kasutades COCO andmestiku objektimaske. Selles töös analüüsitud koodrid on treenitud loomulike piltide ja eri eesmärkidega, nagu näiteks objekti klassifitseerimine (ViT), pildi-keele sobitamine ja pealdise loomine (CLIP, BLIP) ning enesejuhendatud õpe. Me leidsime, et sõltumata treenimise eesmärgist ja arhitektuurist on teatud silmapaistvad objektid teistest paremini esitatud võrgu viimastes kihtides, paljud taustaobjektid aga ei pruugi olla täielikult esitatud märgis, mida kasutatakse järgnevates protsessides (ingl downstream tasks). Selles töös kasutatud sondeerimismeetodit ja leitud tulemusi saab kasutada selleks, et muuta võrgu kihtide ja märkide kasutust järgnevate protsesside ja teiste suuremate mudelite jaoks sobivamaks.

Kokkuvõttes anname selles doktoritöös ülevaate, kuidas mõista nägemist aktiivse informatsiooni kasutamise, näiteks osa-terviku objektihierarhia, kaudu. Selline aktiivne otsing võimaldab kasutada stseenis leiduvaid üksikuid elemente koos suhtestruktuuridega, milles need elemendid tõenäoliselt koos esinevad. Doktoritöö eri osad püüavad tabada seda nähtust, kuidas inimesed ja masinaalgoritmid stseeni tervikuna ja osade kogumina esitavad ning selle üle arutlevad. Lõpetuseks arutleme kolme avaldatud artikli põhjal abstraktsete visuaalsete esituste ning neis toimuva aktiivse otsingu toetavat rolli visuaalses tajus ja stseeni mõistmises.

PUBLICATIONS

CURRICULUM VITAE

Personal data

Full name:	Tarun Khajuria
Date of birth:	24.05.1993
Citizenship:	India
E-mail:	tarunkhajuria42@gmail.com

Education

2020 – 2026	Ph.D. in Computer Science, University of Tartu
2018 – 2020	MSc in Computer Science, University of Tartu
2011 – 2015	B.Tech in IT and Mathematical Innovations, Cluster Innovation Centre, University of Delhi

Employment

2020 –	Junior Research Fellow, University of Tartu
2017 – 2018	Research Associate, IIIT Delhi Innovation and Incubation Centre
2015 – 2017	Co-founder, Imfundo Technologies Pvt Ltd

Supervision

2023, MSc Thesis	Braian Olmiro Dias, (sup) Tarun Khajuria[1] *"Content based analysis of compositionality in vision transformers"*
2022, MSc Thesis	Farid Hasanov, (sup) Jaan Aru[1] , Taavi Luik, Tarun Khajuria *"Iterative Versus Amortized Inference Solutions to the Constellation Problem"*
2022, BSc Thesis	Kalmer Kaurson, (sup) Tarun Khajuria [1] *"Graphical User Interface for Constellations Image Generator"*
2021, Bsc Thesis	Henri Harri Laiho, (sup) Raul Vicente[1], Jaan Aru, Tarun Khajuria *"Recognition as Navigation in Energy-Based Models"*

[1] Supervisor in charge

Teaching

Fall 2025	Natural and Artificial Intelligence Seminar (teaching assistant)
Spring 2020, 2021, 2022, 2023	Neural Networks (teaching assistant)
Fall 2022, 2023, 2024, 2025	Artificial and Natural Intelligence (teaching assistant)

Administrative and professional activities

- Bachelor's and Master's theses defense committee member in June 2025
- Reviewer/ Program Committee: CCN 2023, CVPR 2024,2025, AAAI 2025,2026, IJCV

ELULOOKIRJELDUS

Isikuandmed

Täisnimi:	Tarun Khajuria
Sünniaeg:	24.05.1993
Kodakondsus:	India
E-mail:	tarunkhajuria42@gmail.com

Haridus

2020 – 2026	Ph.D. arvutiteaduses, Tartu Ülikool
2018 – 2020	MSc arvutiteaduses, Tartu Ülikool
2011 – 2015	B.Tech IT ja matemaatilised innovatsioonid, Innovatsiooniklastri keskus, Delhi Ülikool

Teenistuskäik

2020 –	Nooremteadur, Tartu Ülikool
2017 – 2018	Teadur, IIIT Delhi innovatsiooni- ja inkubatsioonikeskus
2015 – 2017	Kaasasutaja, Imfundo Technologies Pvt Ltd

Juhendamine

2023, MSc lõputöö	Braian Olmiro Dias, (sup) Tarun Khajuria[1] *"Sisupõhine analüüs Vision Transformerite kompositsioonilisusest"*
2022, MSc lõputöö	Farid Hasanov, (sup) Jaan Aru[1], Taavi Luik, Tarun Khajuria *"Iteratiivset ja amortiseeritud järeldamist kasutavad lahendused tähtkujude probleemile"*
2022, BSc lõputöö	Kalmer Kaurson, (sup) Tarun Khajuria [1] *"Tähtkujude pildigeneraatori graafiline kasutajaliides"*
2021, Bsc lõputöö	Henri Harri Laiho, (sup) Raul Vicente[1], Jaan Aru, Tarun Khajuria *"Äratundmine navigatsioonina energiapõhistes mudelites"*

Õppetöö

Sügis 2025	Tehisliku ja loomuliku mõistuse seminar (õppeassistent)
Kevad 2020, 2021, 2022, 2023	Tehisnärvivõrgud (õppeassistent)
Sügis 2022, 2023, 2024, 2025	Tehislik ja loomulik mõistus (õppeassistent)

[1] Vastutav juhendaja

Administratiivsed ja professionaalsed tegevused

- Bakalaureuse- ja magistritööde kaitsmiskomisjoni liige juunis 2025
- Retsensent/programmikomitee: CCN 2023, CVPR 2024,2025, AAAI 2025,2026, IJCV

DISSERTATIONES INFORMATICAE PREVIOUSLY PUBLISHED IN DISSERTATIONES MATHEMATICAE UNIVERSITATIS TARTUENSIS

19. **Helger Lipmaa.** Secure and efficient time-stamping systems. Tartu, 1999, 56 p.
22. **Kaili Müürisep.** Eesti keele arvutigrammatika: süntaks. Tartu, 2000, 107 lk.
23. **Varmo Vene.** Categorical programming with inductive and coinductive types. Tartu, 2000, 116 p.
24. **Olga Sokratova.** Ω-rings, their flat and projective acts with some applications. Tartu, 2000, 120 p.
27. **Tiina Puolakainen.** Eesti keele arvutigrammatika: morfoloogiline ühestamine. Tartu, 2001, 138 lk.
29. **Jan Villemson.** Size-efficient interval time stamps. Tartu, 2002, 82 p.
45. **Kristo Heero.** Path planning and learning strategies for mobile robots in dynamic partially unknown environments. Tartu 2006, 123 p.
49. **Härmel Nestra.** Iteratively defined transfinite trace semantics and program slicing with respect to them. Tartu 2006, 116 p.
53. **Marina Issakova.** Solving of linear equations, linear inequalities and systems of linear equations in interactive learning environment. Tartu 2007, 170 p.
55. **Kaarel Kaljurand.** Attempto controlled English as a Semantic Web language. Tartu 2007, 162 p.
56. **Mart Anton.** Mechanical modeling of IPMC actuators at large deformations. Tartu 2008, 123 p.
59. **Reimo Palm.** Numerical Comparison of Regularization Algorithms for Solving Ill-Posed Problems. Tartu 2010, 105 p.
61. **Jüri Reimand.** Functional analysis of gene lists, networks and regulatory systems. Tartu 2010, 153 p.
62. **Ahti Peder.** Superpositional Graphs and Finding the Description of Structure by Counting Method. Tartu 2010, 87 p.
64. **Vesal Vojdani.** Static Data Race Analysis of Heap-Manipulating C Programs. Tartu 2010, 137 p.
66. **Mark Fišel.** Optimizing Statistical Machine Translation via Input Modification. Tartu 2011, 104 p.
67. **Margus Niitsoo**. Black-box Oracle Separation Techniques with Applications in Time-stamping. Tartu 2011, 174 p.
71. **Siim Karus.** Maintainability of XML Transformations. Tartu 2011, 142 p.
72. **Margus Treumuth.** A Framework for Asynchronous Dialogue Systems: Concepts, Issues and Design Aspects. Tartu 2011, 95 p.
73. **Dmitri Lepp.** Solving simplification problems in the domain of exponents, monomials and polynomials in interactive learning environment T-algebra. Tartu 2011, 202 p.

74. **Meelis Kull.** Statistical enrichment analysis in algorithms for studying gene regulation. Tartu 2011, 151 p.
77. **Bingsheng Zhang.** Efficient cryptographic protocols for secure and private remote databases. Tartu 2011, 206 p.
78. **Reina Uba.** Merging business process models. Tartu 2011, 166 p.
79. **Uuno Puus.** Structural performance as a success factor in software development projects – Estonian experience. Tartu 2012, 106 p.
81. **Georg Singer.** Web search engines and complex information needs. Tartu 2012, 218 p.
83. **Dan Bogdanov.** Sharemind: programmable secure computations with practical applications. Tartu 2013, 191 p.
84. **Jevgeni Kabanov.** Towards a more productive Java EE ecosystem. Tartu 2013, 151 p.
87. **Margus Freudenthal.** Simpl: A toolkit for Domain-Specific Language development in enterprise information systems. Tartu, 2013, 151 p.
90. **Raivo Kolde.** Methods for re-using public gene expression data. Tartu, 2014, 121 p.
91. **Vladimir Šor.** Statistical Approach for Memory Leak Detection in Java Applications. Tartu, 2014, 155 p.
92. **Naved Ahmed.** Deriving Security Requirements from Business Process Models. Tartu, 2014, 171 p.
94. **Liina Kamm.** Privacy-preserving statistical analysis using secure multi-party computation. Tartu, 2015, 201 p.
100. **Abel Armas Cervantes.** Diagnosing Behavioral Differences between Business Process Models. Tartu, 2015, 193 p.
101. **Fredrik Milani.** On Sub-Processes, Process Variation and their Interplay: An Integrated Divide-and-Conquer Method for Modeling Business Processes with Variation. Tartu, 2015, 164 p.
102. **Huber Raul Flores Macario.** Service-Oriented and Evidence-aware Mobile Cloud Computing. Tartu, 2015, 163 p.
103. **Tauno Metsalu.** Statistical analysis of multivariate data in bioinformatics. Tartu, 2016, 197 p.
104. **Riivo Talviste.** Applying Secure Multi-party Computation in Practice. Tartu, 2016, 144 p.
108. **Siim Orasmaa.** Explorations of the Problem of Broad-coverage and General Domain Event Analysis: The Estonian Experience. Tartu, 2016, 186 p.
109. **Prastudy Mungkas Fauzi.** Efficient Non-interactive Zero-knowledge Protocols in the CRS Model. Tartu, 2017, 193 p.
110. **Pelle Jakovits.** Adapting Scientific Computing Algorithms to Distributed Computing Frameworks. Tartu, 2017, 168 p.
111. **Anna Leontjeva.** Using Generative Models to Combine Static and Sequential Features for Classification. Tartu, 2017, 167 p.
112. **Mozhgan Pourmoradnasseri.** Some Problems Related to Extensions of Polytopes. Tartu, 2017, 168 p.

113. **Jaak Randmets.** Programming Languages for Secure Multi-party Computation Application Development. Tartu, 2017, 172 p.
114. **Alisa Pankova.** Efficient Multiparty Computation Secure against Covert and Active Adversaries. Tartu, 2017, 316 p.
116. **Toomas Saarsen.** On the Structure and Use of Process Models and Their Interplay. Tartu, 2017, 123 p.
121. **Kristjan Korjus.** Analyzing EEG Data and Improving Data Partitioning for Machine Learning Algorithms. Tartu, 2017, 106 p.
122. **Eno Tõnisson.** Differences between Expected Answers and the Answers Offered by Computer Algebra Systems to School Mathematics Equations. Tartu, 2017, 195 p.

DISSERTATIONES INFORMATICAE UNIVERSITATIS TARTUENSIS

1. **Abdullah Makkeh**. Applications of Optimization in Some Complex Systems. Tartu 2018, 179 p.
2. **Riivo Kikas**. Analysis of Issue and Dependency Management in Open-Source Software Projects. Tartu 2018, 115 p.
3. **Ehsan Ebrahimi**. Post-Quantum Security in the Presence of Superposition Queries. Tartu 2018, 200 p.
4. **Ilya Verenich**. Explainable Predictive Monitoring of Temporal Measures of Business Processes. Tartu 2019, 151 p.
5. **Yauhen Yakimenka**. Failure Structures of Message-Passing Algorithms in Erasure Decoding and Compressed Sensing. Tartu 2019, 134 p.
6. **Irene Teinemaa**. Predictive and Prescriptive Monitoring of Business Process Outcomes. Tartu 2019, 196 p.
7. **Mohan Liyanage.** A Framework for Mobile Web of Things. Tartu 2019, 131 p.
8. **Toomas Krips.** Improving performance of secure real-number operations. Tartu 2019, 146 p.
9. **Vijayachitra Modhukur.** Profiling of DNA methylation patterns as biomarkers of human disease. Tartu 2019, 134 p.
10. **Elena Sügis.** Integration Methods for Heterogeneous Biological Data. Tartu 2019, 250 p.
11. **Tõnis Tasa.** Bioinformatics Approaches in Personalised Pharmacotherapy. Tartu 2019, 150 p.
12. **Sulev Reisberg.** Developing Computational Solutions for Personalized Medicine. Tartu 2019, 126 p.
13. **Huishi Yin.** Using a Kano-like Model to Facilitate Open Innovation in Requirements Engineering. Tartu 2019, 129 p.
14. **Faiz Ali Shah.** Extracting Information from App Reviews to Facilitate Software Development Activities. Tartu 2020, 149 p.
15. **Adriano Augusto**. Accurate and Efficient Discovery of Process Models from Event Logs. Tartu 2020, 194 p.
16. **Karim Baghery.** Reducing Trust and Improving Security in zk-SNARKs and Commitments. Tartu 2020, 245 p.
17. **Behzad Abdolmaleki.** On Succinct Non-Interactive Zero-Knowledge Protocols Under Weaker Trust Assumptions. Tartu 2020, 209 p.
18. **Janno Siim.** Non-Interactive Shuffle Arguments. Tartu 2020, 154 p.
19. **Ilya Kuzovkin.** Understanding Information Processing in Human Brain by Interpreting Machine Learning Models. Tartu 2020, 149 p.
20. **Orlenys López Pintado.** Collaborative Business Process Execution on the Blockchain: The Caterpillar System. Tartu 2020, 170 p.
21. **Ardi Tampuu.** Neural Networks for Analyzing Biological Data. Tartu 2020, 152 p.

22. **Madis Vasser.** Testing a Computational Theory of Brain Functioning with Virtual Reality. Tartu 2020, 106 p.
23. **Ljubov Jaanuska.** Haar Wavelet Method for Vibration Analysis of Beams and Parameter Quantification. Tartu 2021, 192 p.
24. **Arnis Parsovs.** Estonian Electronic Identity Card and its Security Challenges. Tartu 2021, 214 p.
25. **Kaido Lepik.** Inferring causality between transcriptome and complex traits. Tartu 2021, 224 p.
26. **Tauno Palts.** A Model for Assessing Computational Thinking Skills. Tartu 2021, 134 p.
27. **Liis Kolberg.** Developing and applying bioinformatics tools for gene expression data interpretation. Tartu 2021, 195 p.
28. **Dmytro Fishman.** Developing a data analysis pipeline for automated protein profiling in immunology. Tartu 2021, 155 p.
29. **Ivo Kubjas.** Algebraic Approaches to Problems Arising in Decentralized Systems. Tartu 2021, 120 p.
30. **Hina Anwar.** Towards Greener Software Engineering Using Software Analytics. Tartu 2021, 186 p.
31. **Veronika Plotnikova.** FIN-DM: A Data Mining Process for the Financial Services. Tartu 2021, 197 p.
32. **Manuel Camargo.** Automated Discovery of Business Process Simulation Models From Event Logs: A Hybrid Process Mining and Deep Learning Approach. Tartu 2021, 130 p.
33. **Volodymyr Leno.** Robotic Process Mining: Accelerating the Adoption of Robotic Process Automation. Tartu 2021, 119 p.
34. **Kristjan Krips.** Privacy and Coercion-Resistance in Voting. Tartu 2022, 173 p.
35. **Elizaveta Yankovskaya.** Quality Estimation through Attention. Tartu 2022, 115 p.
36. **Mubashar Iqbal.** Reference Framework for Managing Security Risks Using Blockchain. Tartu 2022, 203 p.
37. **Jakob Mass.** Process Management for Internet of Mobile Things. Tartu 2022, 151 p.
38. **Gamal Elkoumy.** Privacy-Enhancing Technologies for Business Process Mining. Tartu 2022, 135 p.
39. **Lidia Feklistova.** Learners of an Introductory Programming MOOC: Background Variables, Engagement Patterns and Performance. Tartu 2022, 151 p.
40. **Mohamed Ragab.** Bench-Ranking: A Prescriptive Analysis Approach for Large Knowledge Graphs Query Workloads. Tartu 2022, 158 p.
41. **Mohammad Anagreh.** Privacy-Preserving Parallel Computations for Graph Problems. Tartu 2023, 181 p.
42. **Rahul Goel.** Mining Social Well-being Using Mobile Data. Tartu 2023, 104 p.

43. **Anti Ingel.** Algorithms using information theory: classification in brain-computer interfaces and characterising reinforcement-learning agents. Tartu 2023, 142 p.
44. **Shakshi Sharma.** Fighting Misinformation in the Digital Age: A Comprehensive Strategy for Characterizing, Identifying, and Mitigating Misinformation on Online Social Media Platforms. Tartu 2023, 158 p.
45. **Kristiina Rahkema.** Quality Analysis of iOS Applications with Focus on Maintainability and Security Aspects. Tartu 2023, 182 p.
46. **Ivan Slobozhan.** Studying Online Social Media Engagement in CIS Countries during Protests, Mass Demonstrations and War. Tartu 2023, 81 p.
47. **Nurlan Kerimov.** Building a catalogue of molecular quantitative trait loci to interpret complex trait associations. Tartu 2023, 248 p.
48. **Pavlo Tertychnyi.** Machine Learning Methods for Anti-Money Laundering Monitoring. Tartu 2023, 117 p.
49. **Abasi-amefon Obot Affia.** A Framework and Teaching Approach for IoT Security Risk Management. Tartu 2023, 180 p.
50. **Raimond-Hendrik Tunnel.** Video Game Design and Development Bachelor's Curriculum for Estonia. Tartu 2024, 137 p.
51. **Ahto Salumets.** Bioinformatics analysis of various aspects in immunology. Tartu 2024, 198 p.
52. **Mohammed Abdulhameed Shaif Ali.** Deep Learning Methods for Cell Microscopy Image Analysis. Tartu 2024, 143 p.
53. **Pille Pullonen-Raudvere**. Foundations of Efficient and Secure Algorithm Development for Secure Multiparty Computation. Tartu 2024, 265 p.
54. **Marili Rõõm.** Multiple approaches to learners' success and factors affecting it in computer programming MOOCs. Tartu 2024, 170 p.
55. **Shivananda Rangappa Poojara.** Design and Orchestration of Scalable, Event-Driven Serverless Data Pipelines for Internet of Things (IoT) Applications. Tartu 2024, 172 p.
56. **Hassan Abdulgaleel Hassan Salim Eldeeb.** Empowering Machine Learning Pipelines with Automated Feature Engineering. Tartu 2024, 121 p.
57. **Muhammad Uzair.** Soft decision making for agri-food 4.0. Tartu 2024, 158 p.
58. **Kirill Milintsevich.** Estimation of Depression Level from Text: Symptom-Based Approach, External Knowledge, Dataset Validity. Tartu 2024, 130 p.
59. **Maksym Del.** Multilingual and Multi-Domain Representational Patterns Across Trpansformer-Based Models. Tartu 2024, 131 p.
60. **Kristo Raun.** Adaptive Out-of-order Handling in Streaming Conformance Checking. Tartu 2024, 118 p.
61. **Toivo Vajakas.** Towards integration of mobile network data into analyzing human mobility. Tartu 2024, 103 p.
62. **Katsiaryna Lashkevich.** Data-Driven Analysis and Optimization of Waiting Times in Business Processes. Tartu 2024, 169 p.
63. **Alejandra Duque-Torres.** Classifying, Constraining and Ranking Metamorphic Relations. Tartu 2025, 159 p.

64. **Mariia Bakhtina.** A Method for Information Security and Privacy Management in Smart Solutions. Tartu 2025, 199 p.
65. **Andre Tättar.** Multilingual Machine Translation for Under-Resourced Languages. Tartu 2025, 170 p.
66. **Mahmoud Shoush.** Prescriptive Process Monitoring Under Uncertainty and Resource Constraints. Tartu 2025, 178 p.
67. **Alireza Akhavi Zadegan.** A Multimodal approach for refining Mapping and Localization by Integrating Generative AI and Pedestrian-Centric Data. Tartu 2025, 147 p.
68. **Eerik Muuli.** Automating the assessment and feedback processes in IT teaching – improving creation and maintenance from the teaching staff perspective. Tartu 2025, 196 p.
69. **Kateryna Kubrak.** Towards User-Centered Prescriptive Process Monitoring Systems. Tartu 2025, 151 p.
70. **Zhigang Yin.** Computing and Sensing in a Smart Ring. Tartu 2025, 251 p.
71. **Abdul-Rasheed Olatunji Ottun.** Practical Trustworthy Artificial Intelligence with Human Oversight. Tartu 2025, 239 p.
72. **Sander Mikelsaar.** Analysis and Optimization of Iteratively Decodable Codes. Tartu 2025, 146 p.
73. **Marharyta Domnich.** Advancing Human-Centric Counterfactual Explanations in Explainable AI. Tartu 2025, 210 p.
74. **Viacheslav Komisarenko.** Aligning Training Loss to Evaluation Metrics in Deep Learning. Tartu 2026, 165 p.
75. **Heidi Taveter.** Using Programming-Process Data of Introductory Programming Courses: Finding Solver Types, Giving Feedback, and Detecting Plagiarism. Tartu 2026, 184 p.
76. **Daniel Majoral Lopez.** Deep neural networks for microscopy images. Tartu 2026, 81 p.
77. **Mahir Gulzar.** Addressing Real-world Scenarios via Motion Prediction in Autonomous Driving. Tartu 2026, 141 p.
78. **Hele-Andra Kuulmets.** Cross-Lingual Transfer Learning and Evaluation in Low-Resource Settings. Tartu 2026, 180 p.
79. **Simmo Saan.** Correctness Witnesses for Thread-Modular Program Analysis. Tartu 2026, 251 p.