

Disassociating Performance from Compositional Feature Learning

George Dimitriadis, Spyridon Samothrakis

Abstract—Out-of-distribution (OOD) generalisation through composition requires a system to discover invariant properties from input–output associations and transfer them to novel inputs and unseen tasks. We argue that confirming compositional learning requires more than OOD evaluation alone: one must also verify that the learned features are genuinely compositional and that the system encodes their compositional rules. We demonstrate this through two tasks with clearly defined OOD metrics, generated via composable high-level abstractions, on which three standard architectures (MLP, CNN, Transformer) and an object-centric, slot-based architecture fail to generalise OOD. We pair these tasks with two novel attention-based architectures featuring an interpretable final hidden layer designed to expose whether compositional representations emerge. One architecture carries an engineered inductive bias that enables near-perfect OOD performance on one task. Our results show that even with appropriate biases and near-perfect OOD accuracy, a model can fail to learn the compositional feature structures necessary for systematic generalisation. The interpretable layer reveals that successful OOD performance is driven by task-specific biases rather than the discovery of reusable compositional primitives. These findings indicate that OOD benchmarks alone are insufficient for evaluating compositionality in neural networks.

Index Terms—compositionality, out-of-distribution, abstract reasoning competition

I. INTRODUCTION

A prominent goal of AI research is to develop machines with human-level cognitive capabilities [1]. Despite extraordinary progress over the last few decades, human and animal cognition still exhibits attributes that elude the field’s state-of-the-art models. Modern AI lacks, for example, the ability of humans and animals to accurately metacognise [2], to transfer knowledge to new problems with vastly different input and output distributions [3], and to generate causal structures from very few examples [4]. It is widely held [5]–[9] that a fundamental property underlying these capabilities is *compositionality*: the ability to extract higher-order representations (concepts) and high-level compositional functions (rules) from input data, and to recombine them to solve out-of-distribution (OOD) problems and reason about counterfactuals.

As the field transitioned from innately compositional symbolic methods to deep learning [10], the question of how to reintroduce compositionality without sacrificing the strengths of neural methods has driven a rich body of work. This work spans neurosymbolic AI [11], probabilistic program induction [1], [12], modular deep neural networks [9], disentangled representation learning [13], object-centric learning [14], [15],

and chain-of-thought reasoning [16]. Nevertheless, productivity and systematicity—two core aspects of compositionality identified by Hupkes et al. [17]—remain largely unattained by current architectures, lending renewed force to Fodor and Pylyshyn’s classical compositionality challenge [10], [18].

In this work, we focus on systematicity: the recombination of known parts and rules to understand novel compositions [17], viewed through the lens of functional compositionality as formalised by van Gelder [19]. We argue that a model exhibiting genuine systematicity will display two characteristics. First, it will demonstrate OOD generalisation across a range of tasks, not only the single task on which it was designed to be evaluated. Second, its latent representation space will exhibit a structure akin to symbolic coding rather than statistical learning: a small (but not overly small) number of features encoding addressable, composition-amenable symbols, rather than a large number of features interacting through weak correlations (see Fig. 1).

Building on theoretical work by Thorpe [20] and the recent complexity-based framework of Elmoznino et al. [21], we argue that many results in the literature on compositional learning (e.g., [22]), although seemingly OOD, do not give rise to such latent features. Their success can be attributed to the model learning the distribution of the task or meta-task, a feat that does not require compositionality. We support this argument by demonstrating how modern architectures can fail to discover the latent feature spaces required for compositional OOD, and how architectures that exhibit task-specific OOD generalisation can do so without employing compositional strategies. We are aware of the large number of specialised architectures that solve tasks through compositional representation discovery (see Section II-B for a non-comprehensive list). However, in their overwhelming majority, such networks are engineered with appropriate biases to achieve compositionality on their target tasks. We contend that in such efforts the hard problem is solved by human engineers who embed the necessary biases into their networks, or through careful design of the task space that gives models only the appearance of having learned compositional latent structure supporting OOD generalisation. Our thesis targets networks that lack explicit compositionality biases and are presumed capable of discovering compositional representations on arbitrary tasks—much like human and animal brains—simply because they exhibit OOD generalisation on a small number of benchmarks.

To make this case, we construct two data sets arising from two world models, based on the input–output design of the Abstract and Reasoning Corpus (ARC) competition [23], [24]. Our tasks are built on parametrised objects with fully defined composition rules, where objects and transformations are separated at the level of the task input, providing the learning algorithm with a disentangled input representation.

George Dimitriadis is with the Computer Science and Electronic Engineering Department of the University of Essex. email: gd25222@essex.ac.uk

Spyridon Samothrakis is with the Computer Science and Electronic Engineering Department of the University of Essex. email: ssamot@essex.ac.uk

The article has code at https://github.com/georgedimitriadis/latent_ood_in_world_models/tree/release_v2

A visual rather than a token-sequence paradigm (such as the SCAN benchmark [25]) was chosen for three reasons. First, newly emerging differences between human and AI visual task solutions [26] suggest that AI algorithms, despite high accuracy scores, do not follow compositional strategies. Second, we sought to avoid certain assumptions about syntactic rules that are often implicit in sequence tasks, as discussed by van Gelder [19]. Third, work with similar goals and conclusions has already been conducted on sequence tasks by Dasgupta et al. [27], who found that models trained on simple sequence tasks learn non-compositional heuristics rather than compositional representations.

Our world models follow the structure presented in [28], which yields a clear definition of OOD through a distance measure between test and training sets. We extend this structure by adding two variables, one semantic (object size) and one geometric (object initial position). These data sets are designed to be shallow in their compositionality depth (detect a single object, find a single rule, apply the rule to the object), a simplicity that affords an intuitive understanding of what compositionality and OOD generalisation should look like. Despite this apparent simplicity, our tasks demand not just a semantic compositional capability (e.g., change colour then change shape) but one that survives geometric changes (e.g., moving or rotating an object between positions never seen during training). In this respect, they add the rule in the compositional chain, demanding from the benchmarked algorithms to compose not only objects and object attributes with each other but also object transformations (actions) with objects. Finally, they allow a discrete measure of whether an algorithm correctly encodes an object and a transformation rule (much like the ARC tasks from which they are derived). Because of this discrete nature, our world models admit as a test metric not a continuous, fuzzy measure (such as mean squared error), where meaningful errors can be masked by averaging over pixels, but a discrete one (correct image or not), where only a true conceptualisation of the objects and their one-step transformation rules can yield a correct solution.

We evaluate these data sets on three groups of architectures. The first group comprises three commonly used architectures—an MLP, a CNN, and a Transformer—none of which is known for solving problems compositionally. These serve as baselines that compositional models should be expected to surpass. The second group consists of a single architecture with an object-centric inductive bias, the Object-Centric Learning (OCL) model of [29], designed to detect objects in images and compose object-manipulation rules over them. OCL is a slot-based architecture currently considered state-of-the-art (SOTA) among object-centric, compositionally based algorithms. The third group consists of two architectures of our own design. These share three key properties: they achieve substantially better OOD performance on our data sets than both the standard architectures and the object-centric one; they possess an interpretable final latent layer that can be coherently visualised, enabling unambiguous detection of whether compositional representations have been learned; and the two architectures differ only in that one carries an additional inductive bias that makes one of the two data sets more

OOD-learnable, thereby clearly illustrating how an engineered rather than learned bias can be the sole driver of apparent OOD behaviour while degrading performance on tasks whose solution does not align with it.

We propose that evaluating algorithms as “compositional” based solely on their results on specific benchmarks can often be misleading, especially when the task/metric combination can obscure the true nature of an algorithm’s learned representations, even when those results indicate a high degree of OOD learning. For AI to achieve compositionality, a more nuanced evaluation is required—one that characterises new algorithms both with a diverse set of OOD benchmarks and with methods that directly probe the structure of their latent feature spaces.

The remainder of this paper is organised as follows. Section II reviews the concept of compositionality and relevant prior work. Section III describes the methodologies used for generating the data sets and the models evaluated. Section IV presents our results, demonstrating a clear case in which models that achieve high OOD accuracy (given appropriate biases) still fail to develop latent features suitable for compositional generalisation over feature-invariant tasks. Section V discusses the main limitations and avenues for future work.

II. BACKGROUND

A. Compositionality: An Overview

Compositionality was originally introduced as a key aspect of human language [6], [30], [31] and as a common property of language and the way minds understand the world [32]. In linguistics, compositionality is defined as “The meaning of a whole is a function of the meanings of the parts and of the way they are syntactically combined” [33]. In the behavioural sciences, compositionality refers to “hierarchical nesting of parts into larger wholes with new individuality, identity, or meaning—or inversely, partitioning of such wholes into smaller parts with independence or individual meaning of their own” [7]. A substantial body of work has further shown that not only humans but also animals approach the decomposition of sensory input and the composition of actions in a compositional fashion [7], [34], [35].

The observation that nature appears compositional in structure motivates the assumption that cognition achieves two goals. The first is to extract from input data higher-order representations (concepts) as well as high-level compositional functions (rules) that operate on both concepts and on themselves (see [19] for a definition of composable concepts and rules; see also [36] for a counterposition on humans’ ability to extract concepts and rules from data). The second is to use those abstractions to solve OOD problems and generate possible solutions to counterfactuals. In humans, such abstractions bind together and render accessible large swathes of sensory and action spaces, most of which are inexperienced or even physically impossible [36], [37].

B. Modern Approaches to Compositionality in Machine Learning

As the field of AI transitioned from symbolic methods to connectionist and deep learning approaches [10], the argument

for the importance of compositionality remained strong [18]. Van Gelder placed the concept of compositionality within connectionist approaches on a firm theoretical basis [19]. From that point, a proliferation of methodologies has followed, including neurosymbolic AI [11], probabilistic program induction [1], [12], modular deep neural networks [9], disentangled representation learning [13], object-centric learning [14], [15], [29], and chain-of-thought reasoning [16]. For reviews, see also [38], [39], and [10].

Hupkes et al. [17] addressed the application of the traditional linguistic notion of compositionality within a modern machine learning context. They decomposed the concept into five underlying notions: systematicity, productivity, substitutivity, localism, and overgeneralisation. They then constructed theoretically grounded compositionality benchmarks for three commonly used neural network architectures: Long Short-Term Memory networks (LSTM) [40], Convolutional Networks (CNN) [41], and Transformers [42]. Their findings indicated that productivity and systematicity remain unattainable by current models, with Transformers performing better than the other two architectures. These results have renewed attention to Fodor and Pylyshyn’s argument [18] regarding the compositional deficiency of connectionist architectures, also known as the compositionality challenge of neural networks [10]. One of the most recent efforts towards addressing this challenge was the meta-learning-based approach of Lake and Baroni [22], who showed that on a series of linguistic tasks, an attention-based model could learn novel, unseen tasks (automatically generated by a pre-specified process) through the use of only a few in-context examples. In the field of image recognition and processing, object-centric, slot-attention-based algorithms such as the OCL architecture of Dedhia et al. [29] (a combination of the SLATE algorithm [43] and an attention-based analogy-discovery process) are purported to appropriately encode object-centric structures and composable object transformations.

C. Systematicity in Representation

In this work, we focus exclusively on systematicity as defined by Hupkes et al.: “This ability concerns the recombination of known parts and rules: anyone who understands a number of complex expressions also understands other complex expressions that can be built up from the constituents and syntactical rules employed in familiar expressions.” from the viewpoint of functional compositionality as proposed by van Gelder. Functional compositionality does not assume concatenation-based syntactic rules (that compose tokens into larger expressions) and is a form of compositionality more consistent with connectionist approaches [19].

We argue that a model exhibiting systematicity will display two characteristics. First, it will demonstrate OOD generalisation across a range of tasks (as do animals and humans), and not only on the single task on which it was designed to be evaluated. Second, its latent representation space (if appropriately probed) will exhibit a structure resembling symbolic coding rather than statistical learning. Specifically, there will be a small (but not overly small) number of features encoding

addressable and composition-amenable symbols, rather than a large number of features interacting through an even larger number of weak correlations (see Fig. 1 for a schematic illustration). This characteristic derives from the work of Thorpe [20] (see also the analysis by Olah [44]) and, more recently, Elmoznino et al. [21]. Thorpe’s work establishes that composable representations require a certain degree of compressibility between one-to-one mapping and maximally compressed coding. Elmoznino et al. place this intuition on a theoretical basis, arguing that a compositional representation is a semantic mapping from a set of sentences composed of symbolic features (tokens) to a set of continuous vectors, with the tokens, their syntax, and the semantic map chosen such that the representation is maximally compressed in the Kolmogorov complexity sense. Their work highlights the difficulty of discovering such a language (tokens and their syntactic rules) and then representing it in a model’s continuous latent features.

We argue that many results in the literature on compositional learning (e.g., [22]), although seemingly OOD, do not give rise to the required latent features. Their success on the evaluated tasks is simply the result of the model learning the distribution of the task or meta-task (i.e., the set of all outputs the data-generating process can produce), a feat that does not require compositionality.

III. METHODS

A. World Models and Data Sets

To validate our thesis, we construct two data sets derived from two distinct world models. These share identical input and output observation spaces but differ in their action spaces and the corresponding distributions over observables. Both world models operate on images structurally analogous to those in the ARC Competition [24]. Each image consists of a 32×32 pixel grid where each pixel takes one of four colour values. White pixels denote unused space, black pixels define an $n \times m$ background canvas ($n, m \leq 32$), and pixels of the remaining two colours (red and blue) represent objects placed on this canvas. Each input–output pair contains a single coloured object, consistent across both images, defined by a set of coloured pixels. Objects may extend partially beyond the canvas boundary, in which case only the visible portion is rendered.

The design of these two world models follows the compositional structure proposed in [28]. Each model is parameterised by five independent concept variables (CVs) that govern the appearance and transformation of objects within the images. Four of these are binary-valued and one takes values in $\mathbb{N}^{32 \times 32}$. Across both models, the CVs are: *Shape*, *Colour*, *Size*, *Position* on the canvas, and *Action*, which specifies the transformation applied to the object to produce the target output (see Fig. 2A and B for the domain of each CV).

Each world model is named according to its *Action* CV. The **Translate** model defines actions **Move Left** and **Move Up** (each by 6 pixels), while the **Rotate** model defines actions **Rotate 90** and **Rotate 180**. For each world model, a training set and multiple held-out test sets are generated to evaluate OOD

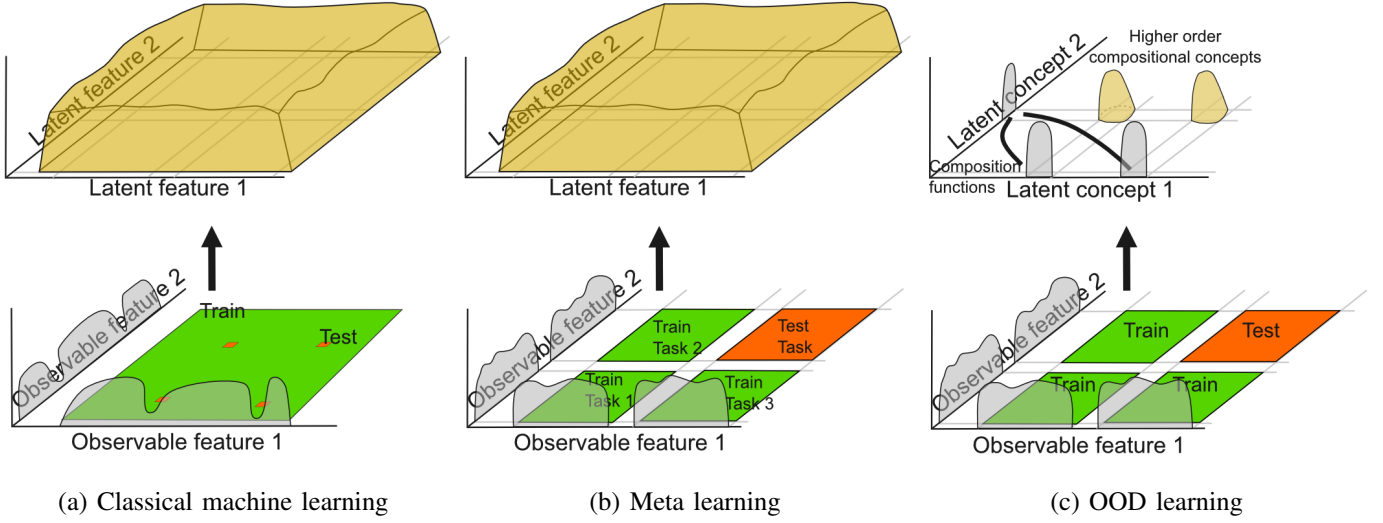


Fig. 1: A conceptual understanding of the relationship between classical machine learning algorithms, modern meta learning ones and a working out of distribution algorithm based on compositionality. We argue that current ML frameworks, both from classical ML and from meta-learning approaches generate latent features in spaces (yellow regions) with no real compositional capability. These representations are spread throughout the latent space and operate on the base of a large number of small correlations between subregions. A true compositional architecture would be able to discover latent features whose distribution over their spaces would appear much more multimodal and discreet (concepts), allowing them to combine those concepts into higher order compositions without modes of failure arising from statistical mixing of broad distributions.

generalisation. All possible combinations of three CVs (*Shape*, *Colour*, *Action*) are enumerated, yielding a concept graph of 8 independent combinations (see Fig. 2C). These combinations are partitioned into three concept classes. The class containing four combinations is used to construct the training set and an in-distribution test set (denoted Test Distance 0). The remaining four combinations form two additional OOD test sets: one comprising combinations at graph distance 1 from the training distribution (Test Distance 1), and one comprising the single combination at graph distance 2 (Test Distance 2). The remaining two CVs (*Position* and *Size*), as well as the canvas dimensions, are sampled uniformly at random across all data sets, subject to a minimum canvas size of 10 pixels in each dimension.

Each sample consists of an input tuple paired with a target output image. The input tuple comprises the source image and a binary action indicator specifying the transformation to be applied. As a concrete example from the Translate world model: the input image might depict a large (*Size*) 7×7 pixel blue (*Colour*) square (*Shape*) with its bottom-left pixel at grid position (*Position*) (6, 1) on a 10×10 black canvas. The action indicator (*Action*) is 0 for **Move Left** or 1 for **Move Up**. The corresponding target image contains the same canvas and object, translated by 6 pixels in the direction specified by the action indicator.

B. Algorithms

We evaluate the OOD generalisation capabilities of six architectures on the above data sets: an MLP, a CNN, a Transformer, a slot-attention model with an object-centric inductive bias, and two bespoke attention-based networks

designed to achieve strong OOD performance on one or both world models.

The five architectures other than OCL share a common dual-pathway input structure that merges into a single output pathway (see Fig. 2D). The first pathway processes the input image, with the specific layer types determined by the architecture. The second pathway receives the binary action indicator, which is projected into a learned embedding vector whose dimensionality is architecture-dependent. The image representation and action embedding are concatenated and passed through subsequent architecture-specific layers to produce the output. The OCL model is instead designed to discover the rule from a number of extra image pairs provided as examples (referred to in the original work as analogies); here we used two analogy pairs per sample, drawn from the same distribution defining the class of the sample (Train, or Distance 0, Distance 1, or Distance 2, and Translate or Rotate). To ensure a fair comparison, all models are matched as closely as possible in total parameter count: the Transformer has 4.6M parameters, the CNN 4.5M, the MLP 6.2M, and both axial pointer networks 4.4M. OCL also has a comparable number of parameters (5.5M) (see Table I for details).

The hyperparameters of the OCL model were set to match the configuration of the original publication [29] as closely as possible, using the code from its GitHub repository directly. We used two analogy image pairs per sample (a 2-shot setting), three slots, and a vocabulary size of 128 in the SLATE algorithm, with 30,000 steps for the SLATE pre-training stage and 150 epochs (80,000 training samples and 20,000 validation samples) for OCL training. The remaining hyperparameters follow [29] and [43]. The choice of three slots reflects the three “objects” the algorithm should be able to detect and

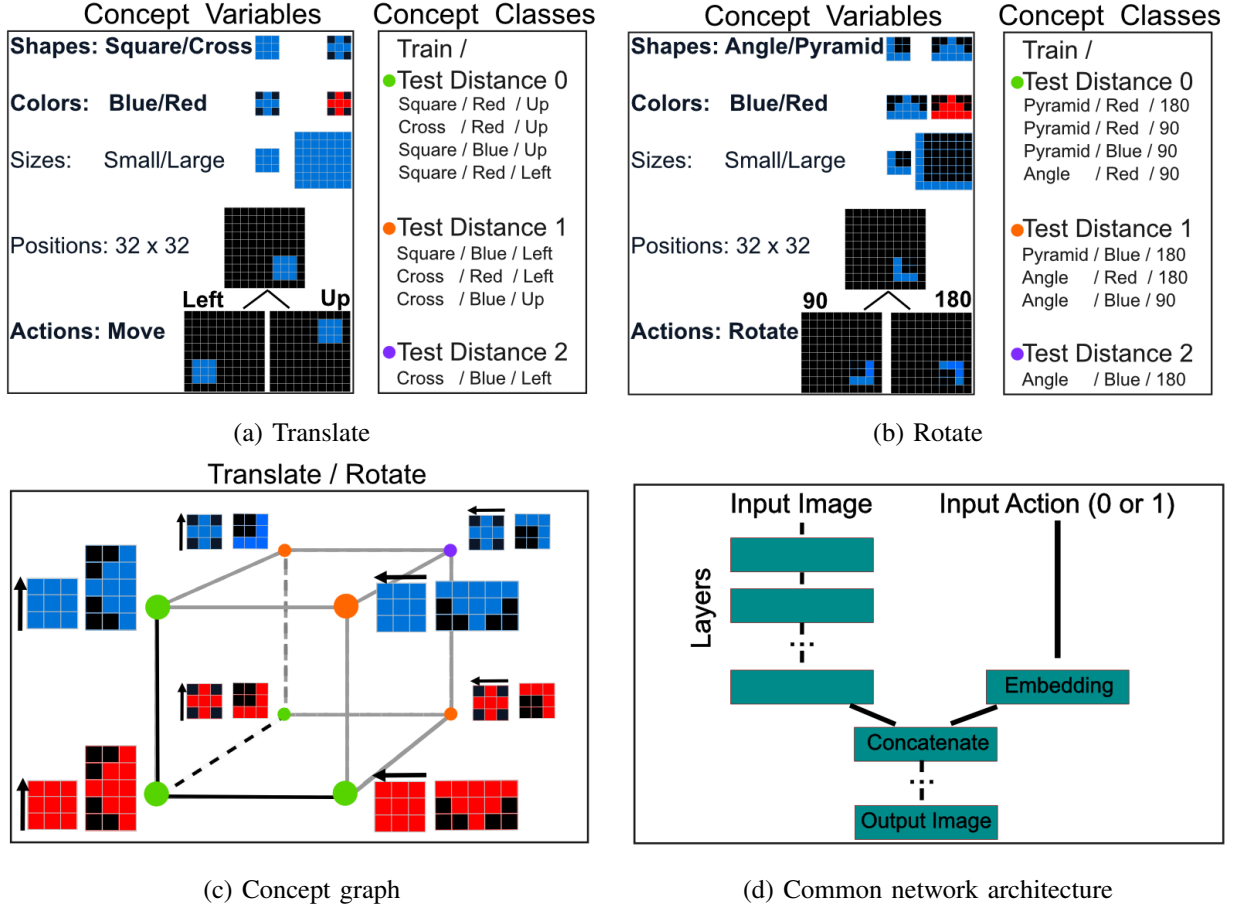


Fig. 2: Overview of our methodology. (a) The design of the Translate world model with its derived data sets. (b) The design of the Rotate world model with its derived data sets. (c) The Concept Graphs for both world models showing which combinations of CVs belong to which data set (also shown in a and b in the Concept Classes boxes) and the distance between each combination on the unit cube. So a Train or a Test Distance 0 sample would be derived from combining the values in the Train / Test Distance 0 column in b), shown diagrammatically as the green dots of the cube in c). For example a possible Train sample would be a blue square **moving Up** or a red pyramid **rotating 90** (top left front green corner of the Concept Graph cube). A Distance 2 sample could only be a blue cross **moving Left** or a blue angle **rotating 180** (seen in the top right back purple corner of the Concepts graph) (d) The common architecture for the algorithms in this work that take as input the images and the language bit denoting the transformation rule. The architecture of the OCL algorithm was identical as the one used in its presenting publication. The type of network defined the left hand path types of layers and the embedding size of the right hand path.

separate: the black background, the white surround, and the coloured object.

The two bespoke architectures incorporate gated MLPs (gMLP) [45], axial attention [46], and pointer networks [47]. The pointer network layer implements an attention mechanism that learns a copy operation from input to output pixels, effectively computing a mapping from each input pixel index to a corresponding output pixel index. This architectural choice yields an interpretable mapping that can be directly inspected to analyse how each model transforms inputs across the different data sets (see Section IV and Fig. 5). The axial attention mechanism, integrated with the pointer attention layer, factorises the 2D index prediction into independent computations along each spatial axis.

Specifically, in the axial pointer network (APN), the x and y components of each pixel's destination index are predicted

independently. We introduce a variant, the axial pointer linear network (APLN), which first predicts the target column index and subsequently predicts the row within that column. In the algorithms panel we show the inference and training loop, common to the two algorithms (Algorithm 2 and Algorithm 3) as well as their diverging part in the training (Algorithm 1). The difference is in the pointer factorisation, i.e. the shape and meaning of the logits the control MLP emits: the APN emits an independent axial pointer per target pixel ($[H, W, H] + [H, W, W]$), while the APLN emits a single global row-mixing matrix and a single column-mixing matrix ($[H, H] + [W, W]$) applied to the whole canvas. The lines variant is therefore a hard architectural prior that the task is a whole-line rearrangement, bought with a $32\times$ smaller output head. This decomposition introduces an inductive bias favouring column- and row-wise copying operations consistent with

Algorithm 1 Axial pointer network forward pass. The trunk is shared; the two variants differ only in the size of the logit head and the granularity of the copy operation.

Require: grid $X \in \{0, \dots, 10\}^{H \times W}$, task index z , mode $\in \{\text{FULL}, \text{LINES}\}$

Ensure: predicted grid $Y' \in [0, 1]^{H \times W \times 11}$

Shared trunk

- 1: $x \leftarrow \text{OneHot}(X, 11)$ $\triangleright$ canvas to be copied from
- 2: $\tilde{x} \leftarrow x + \mathcal{N}(0, 0.05)$ $\triangleright$ Gaussian noise, training only
- 3: $p \leftarrow \text{PatchEncoder}_{256}(\text{Patches}_{8 \times 8}(\tilde{x}))$ $\triangleright$ 16 patches + positional emb.
- 4: **for** $g = 1, \dots, 10$ **do**
- 5: $p \leftarrow \text{gMLP}(p)$ $\triangleright$ spatial-gating mixing
- 6: **end for**
- 7: $f \leftarrow \text{Flatten}(p)$
- 8: $e_z \leftarrow \text{Embedding}(z)$ $\triangleright$ task embedding, 128-d
- 9: $c \leftarrow \text{MLP}_{\text{res}}([f; e_z])$ $\triangleright$ control MLP; 32 units (FULL) / 256 units (LINES)
- 10: $\ell \leftarrow \text{Dense}(c)$ $\triangleright$ attention logits; *size differs by variant*

Divergent trunk - APN

- 11: **if** mode = FULL **then** $\triangleright$ pixel-level pointers,
 $|\ell| = (H+W)HW = 65,536$
- 12: $L_h \leftarrow \text{reshape}(\ell_{[:HWH]}) \in \mathbb{R}^{H \times W \times H}$ $\triangleright$ per target pixel: dist. over source rows
- 13: $L_w \leftarrow \text{reshape}(\ell_{[HWH:]}) \in \mathbb{R}^{H \times W \times W}$ $\triangleright$ per target pixel: dist. over source cols
- 14: $P_h, P_w \leftarrow \text{Point}(L_h), \text{Point}(L_w)$ $\triangleright$ Alg. 2
- 15: $T_{i,w,c} \leftarrow \sum_h (P_h)_{i,w,h} x_{h,w,c}$ $\triangleright$ gather along height
- 16: $Y'_{i,j,c} \leftarrow \sum_w (P_w)_{i,j,w} T_{i,w,c}$ $\triangleright$ then along width
 $\triangleright$ each output pixel (i, j) independently copies from its own chosen (row, col)

Divergent trunk - APLN

- 17: **else if** mode = LINES **then** $\triangleright$ line-level pointers,
 $|\ell| = H^2 + W^2 = 2,048$
- 18: $A_r \leftarrow \text{reshape}(\ell_{[:H^2]}) \in \mathbb{R}^{H \times H}$ $\triangleright$ one global row
 $\rightarrow$ row map
- 19: $A_c \leftarrow \text{reshape}(\ell_{[H^2:]}) \in \mathbb{R}^{W \times W}$ $\triangleright$ one global col
 $\rightarrow$ col map
- 20: $A_r, A_c \leftarrow \text{Point}(A_r), \text{Point}(A_c)$ $\triangleright$ Alg. 2
- 21: $Y' \leftarrow A_r x$ $\triangleright$ move entire rows
- 22: $Y' \leftarrow (A_c Y'^\top)^\top$ $\triangleright$ then entire columns
 $\triangleright$ all pixels of an output row come from one source row (likewise columns): only whole-line moves are expressible
- 23: **end if**
- 24: **return** Y'

translational transformations, and was designed to facilitate OOD generalisation on the Translate data sets while potentially hindering performance on the Rotate data sets. Our experimental results confirm this intended behaviour. Finally, we note that the gMLP layers play a critical role in producing well-calibrated logits for the attention mechanism; replacing

Algorithm 2 Point(L): soft pointer during training, hard pointer at evaluation.

Require: logits L (softmax over the last axis), temperature $\tau = 1$

- 1: **if** training **then**
- 2: $u \sim \mathcal{U}(0, 1)$; $g \leftarrow -\log(-\log u)$ $\triangleright$ Gumbel noise
- 3: **return** softmax($(L + g)/\tau$) $\triangleright$ soft, differentiable
- 4: **else**
- 5: **return** OneHot($\arg \max L$) $\triangleright$ hard, discrete copy
- 6: **end if**

Algorithm 3 Training and per-distance evaluation loop (identical for FULL and LINES).

Require: train set (X, Z, Y) ; test sets $\{(X_d, Z_d, Y_d)\}_{d \in \{0,1,2\}}$ at compositional distance d

- 1: compile with AdamW ($\eta = 10^{-4}$, weight decay 0.004), sparse categorical cross-entropy
- 2: **for** epoch = 1, $\dots$, N **do**
- 3: fit one epoch on $(X, Z) \rightarrow Y$, batch size 100
 $\triangleright$ forward uses noisy input and *soft* Gumbel pointers: gradients flow through the copy op
- 4: **for** $d \in \{0, 1, 2\}$ **do**
- 5: evaluate on (X_d, Z_d, Y_d) and log score $\triangleright$
evaluation uses *hard* arg max pointers: the network must commit to discrete copies
- 6: **end for**
- 7: **if** epoch mod $n_{\text{save}} = 0$ **then** save model (and figures)
- 8: **end if**
- 9: **end for**

them with standard MLP layers substantially degrades OOD generalisation performance (data not shown).

Network	Total Parameters	Trainable Parameters
MLP	6,170,976	6,170,976
CNN	4,474,031	4,466,215
Transformer	4,663,040	4,663,040
OCL	5,525,891	5,525,891
APN	4,481,088	4,480,384
APLN	4,446,112	4,440,480

TABLE I: Number of parameters for each of the six networks used.

IV. RESULTS

A. OOD Generalisation Is Achieved Only by the Appropriately Biased Architecture

Fig. 3(a) and (b) present the learning curves of all architectures across the six test sets (three from each world model). The reported metric is the percentage of fully correct output images produced by each network on each test set. As expected, all architectures perform well on the Distance 0 test sets for both world models, as these are drawn from the same concept classes as the corresponding training sets. The CNN converges to 100% accuracy faster than all other architectures; we hypothesise that its architecture allows it to memorise all

input-action-output combinations more rapidly than the other networks. The OCL architecture reaches only 0.74 accuracy on both the Translate and Rotate world models, the lowest of all architectures. We report the first 100 epochs here, but we ran OCL for a total of 150 epochs without any change in the results.

On the Translate world model, the APLN achieves perfect accuracy within fewer than 10 epochs, while the APN plateaus at approximately 80%. On the Distance 1 test set, the Transformer attains roughly 40% accuracy, whereas the CNN and MLP do not exceed 10%. OCL achieves 1.3% on the Translate Distance 1 test set and 0% on the corresponding Rotate test set. On the Distance 2 test set, the APLN demonstrates full OOD generalisation, again reaching 100%. The APN performs slightly worse on Distance 2 than on Distance 1, indicating that the axial pointer mechanism without the linear decomposition is insufficient for this world model. The remaining four architectures fail to produce a single correct output image on the Distance 2 test set, indicating that they are unable to learn latent representations supporting OOD generalisation.

All architectures were matched in parameter count for fair comparison (see Table I). This required substantially reducing the dimensionality of the final gMLP layer in the APN (from 256 in the APLN to 32 in the APN), as the APN allocates considerably more parameters to the pixel copying layer. This constraint negatively impacts APN performance. Fig. 3(c) presents additional comparisons between the APN and the APLN on the Rotate data sets under relaxed parameter budgets. In two configurations, both networks have approximately 20M parameters (final MLP layer size of 256 for the APN, yielding 20.7M parameters, and 1024 for the APLN, yielding 19.1M). In two further configurations, both networks have approximately 80M parameters (final MLP hidden layer size of 2560 for the APLN and 1024 for the APN). With comparable and substantially larger parameter budgets, the APN exhibits only a marginal improvement over the APLN on the Distance 1 test set. At higher parameter counts, the APN also achieves perfect accuracy on the Distance 0 test set. In the case of OCL, the model already had a comparable number of parameters when used with the hyperparameters suggested by [29].

As described above, the APLN was designed with an inductive bias towards copying pixels in vertical and horizontal groups. The Rotate world model was specifically constructed to test the importance of such a bias for OOD generalisation. As shown in the Rotate panel (b) of Fig. 3, none of the architectures—including the APLN—exhibit any meaningful OOD generalisation on this world model, confirming our hypothesis. Both axial pointer architectures partially solve the Distance 1 test set. The remaining three architectures perform worse than on the Translate world model, with only the Transformer again showing a slight advantage over the CNN and MLP. The Transformer’s relative performance across both world models is consistent with the findings of Hupkes et al., who reported that Transformer architectures achieve modestly better OOD generalisation on systematicity tasks compared to other standard architectures.

B. Error Analysis Reveals Non-Compositional Representations

Examining the errors each architecture produces on the OOD test sets provides insight into whether a model can at least approximate the correct output. Errors that preserve the correct object shape and/or apply the correct transformation suggest that the network has learned, at least partially, a disentangled representation separating object attributes from the required action.

Fig. 4 presents characteristic errors across the Distance 1 and Distance 2 test sets for both world models. Across all cases, the MLP produces outputs with the least resemblance to the ground truth, consistent with the known limitations of MLPs on image recognition and generation tasks. CNN errors approximate the correct output only on the Translate Distance 1 test set. In this case, nearly all CNN errors consist of an incorrect translation of the correct object (data not shown), as illustrated in the figure. This suggests that the network successfully copies the object’s shape and colour from the input but fails to compose these with the correct action. Instead, it applies only the actions observed during training for each object type, having learned latent features that conflate object appearance with the associated translation. On the remaining test sets (Translate Distance 2 and both Rotate sets), CNN errors exhibit the same lack of grounding to objects and actions as those of the MLP.

The Transformer consistently produces errors closer to the ground truth across all test sets, generating approximately correct objects (with minor pixel additions or omissions) and applying approximately correct transformations. On the Distance 2 sets, erroneous pixels may also have incorrect colour values, a phenomenon not observed at Distance 1 (data not shown). These error characteristics are again consistent with the findings of Hupkes et al. regarding the Transformer’s superior OOD performance.

The majority of the OCL network’s mistakes are either in the rule (creating the correct object but not positioning it appropriately) or cases where both the object and the rule are wrong. In a large proportion of the mistaken-object cases (both with and without the correct rule), the network generates one of the other possible objects. For example, in the Rotate/Distance 2 case, instead of a blue angle rotating 180°, the model often generates a 90°-rotated small blue pyramid (data not shown, but see Fig. 4(c)(OCL) for an example). In the remaining cases where the object is wrong, the network misses the correct object by a few pixels (e.g., a truncated pyramid—see Fig. 4(d)(OCL)—or a square with a few missing or extra pixels). Table II gives, for each world model and distance, the percentage of mistakes across four error types: (i) correct shape and rule (final positioning) but wrong colour, (ii) correct rule but wrong shape, (iii) correct shape but wrong rule, and (iv) both shape and rule wrong (for types ii, iii, and iv we ignore colour). These results emphasise that, despite OCL’s bias towards object recognition, the majority of mistakes still involve a combination of misshaped objects and mistaken rule application.

Despite the OCL architecture’s focus on object recogni-

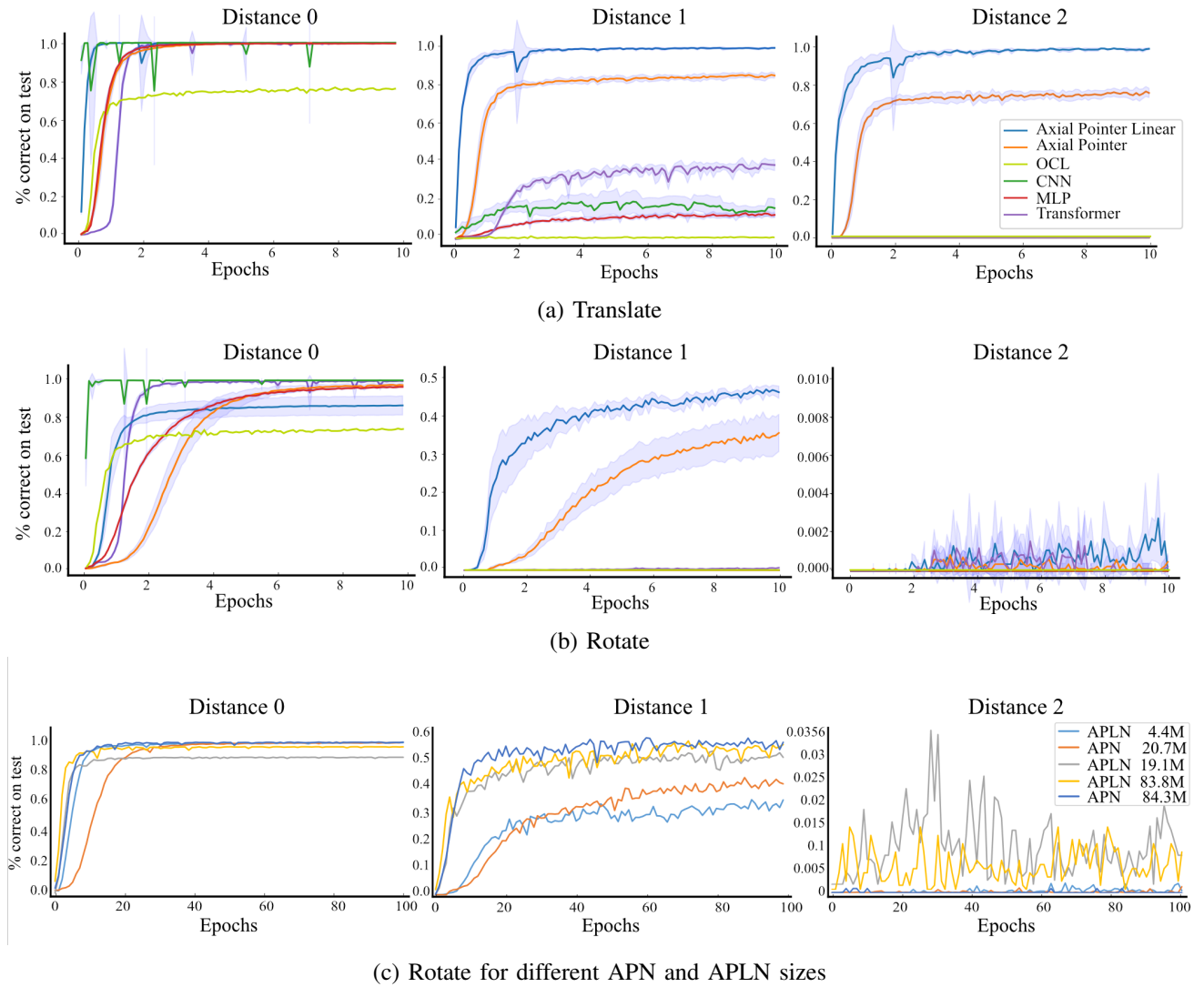


Fig. 3: The results of all networks on the 6 test data sets arising from the two world models. The confidence intervals are generated from 10 runs with a 95% confidence level. The results of all six networks on Translate (a) and Rotate (b) world models where all networks have their total number of parameters matched. (c) The results of the APN and APLN for the Rotate world model showing different versions of the networks with different number of total parameters.

tion, object segmentation, and compositional rule application, the model fails on our world models notwithstanding their compositional simplicity. We propose that our world models, combined with the discrete, full-correct-image metric they allow, are a more appropriate test not just of compositionality but of actual OOD generalisation. As shown by the errors in Fig. 4, the OCL architecture manages to assign only roughly the correct object to the input slots (with a large number of extra pixels and, sometimes, some pixels missing). Despite this, the output generated by the network is, most of the time, a combination of the wrong object and the wrong rule application. The objects finally generated are sometimes objects the network has seen in other samples, and sometimes mangled objects with an obvious resemblance to objects seen in the training set. Our initial hypothesis for this failure mode is that the network is designed to compose over objects and object attributes (assigned to slots) but its attention over slots

composition layer is unable to deal with composition over actions (because the slots do not encode the action space of the world models). It is possible that increasing the number of slots to match the number of possible position/orientation combinations for each object (thus encoding all possible object transformations in this large set of slots) might allow the network to produce the correct object output. Were this confirmed, it would suggest that any resulting improvement stems from memorising the training set rather than from genuine OOD-capable generalisation.

The structure of the training and testing data sets implies an imbalance that could affect the way the networks learn. In the training set, there are three times as many samples of one rule (**Up** and **90**) as of the other. This could bias the networks towards one rule over the other. To better understand the impact of this imbalance, we show in Table III the number of samples across error types for the four rules (**Up**, **Left**,

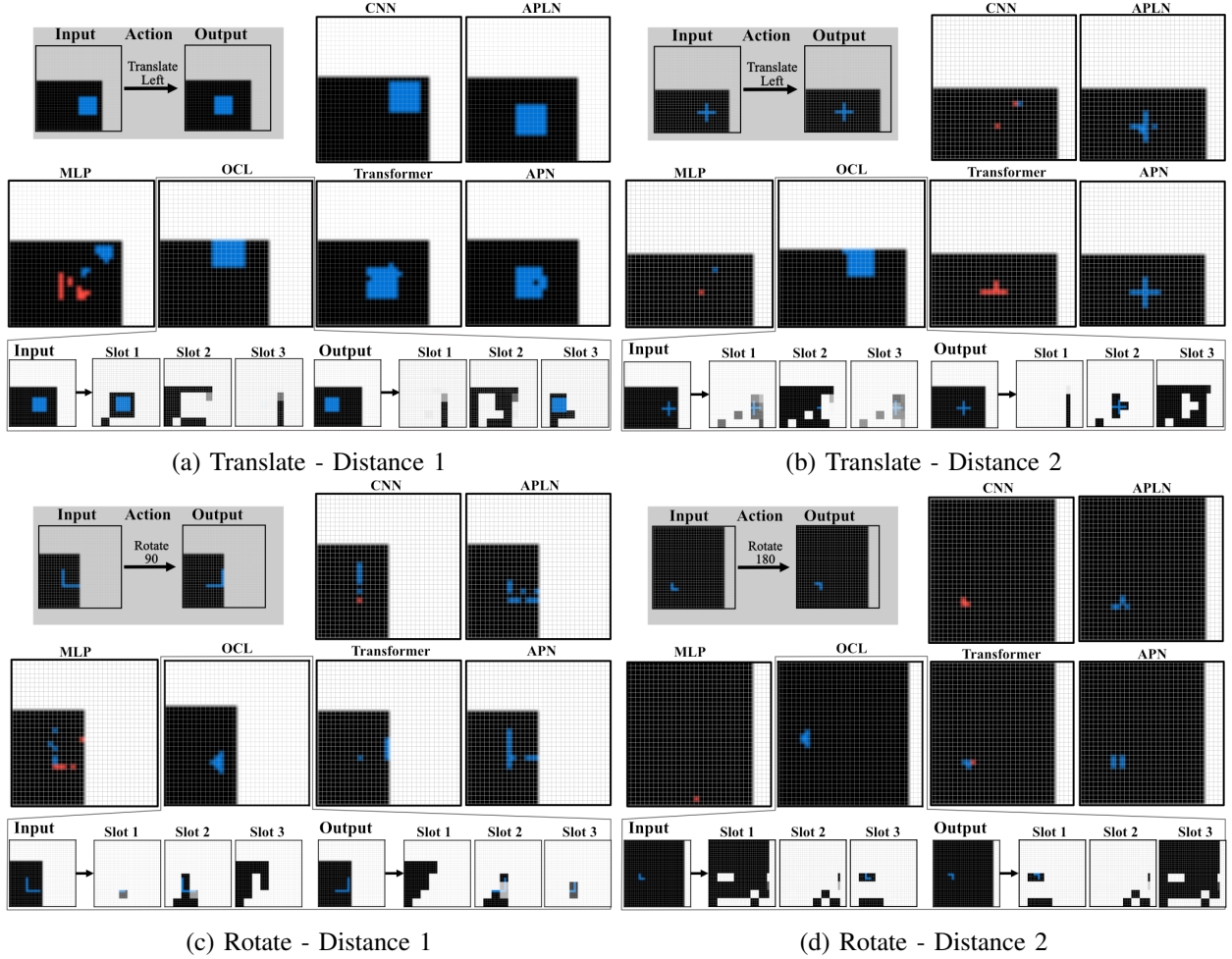


Fig. 4: Characteristic errors that the networks make in the different test sets. In the top left corner of each sub figure (greyed out) we show the ground truth of the specific datum (input image, action bit and output image). In the Translate - Distance 1 test set (A) the APLN was 100% correct so we are showing a correct output image. For the OCL error, the input and output images are shown in the third image row of each sub figure. There we also show the three slots discovered by OCL for each of these input and output images.

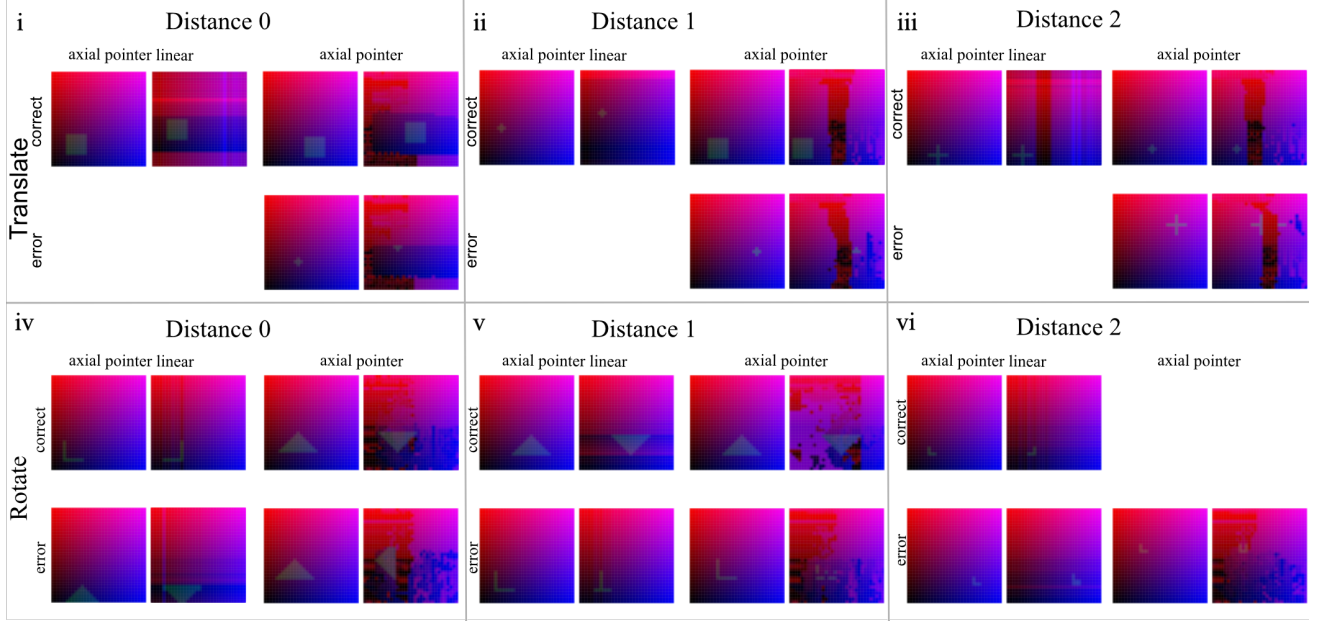
90, and 180). It is evident that this 3:1 imbalance affects the number of fully correct and fully wrong samples depending on the rule. The number of fully correct samples for the **Up** and **90** rules is about three times that for the **Left** and **180** rules (e.g., 532 correct for **Up** vs. 247 for **Left**), while the opposite ratio holds for the all-wrong sample category (e.g., 1321 for **Left** vs. 408 for **Up**). This transfer of the sample-distribution imbalance into the distribution of OCL error types underscores that even networks with object-based inductive biases, presumed to solve their tasks through object and rule composition, can fail to learn in a genuinely OOD-generalisable way and instead remain in the regime of learning irrelevant statistical correlations.

The axial pointer networks produce substantially more accurate outputs on the Translate world model, where errors typically consist of small numbers of omitted or spurious pixels. This is not the case on the Rotate world model, where the pixel copying operates in a near-random fashion, though concentrated in the spatial region where the correct object should appear. As expected, given that the APN and

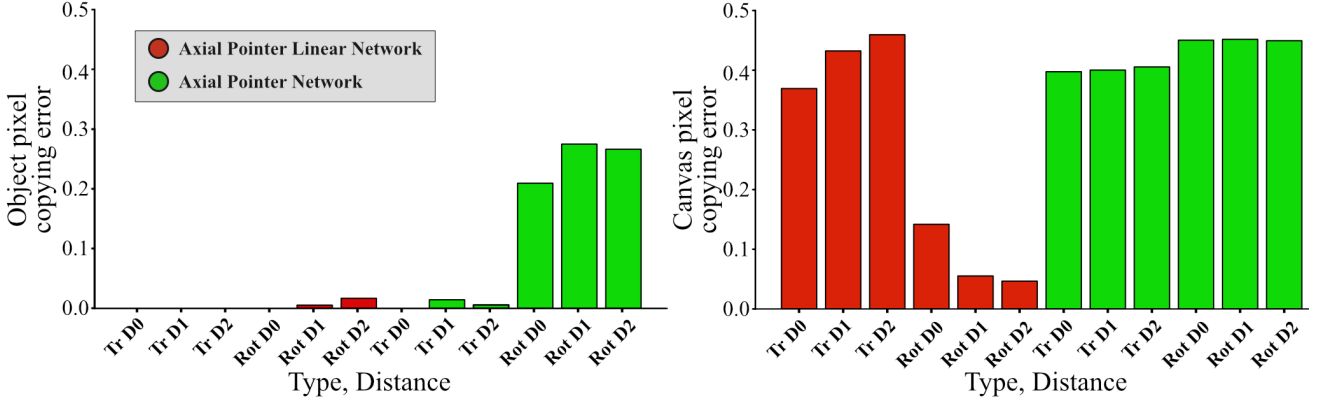
APLN operate by copying pixels from input to output, these architectures never produce pixels of incorrect colour. Overall, the error patterns across all architectures indicate that even in cases where the aggregate error rate is low but non-zero, none of the networks have learned genuinely compositional representations of either the objects or the required transformations.

Error type	T/D0	T/D1	T/D2	R/D0	R/D1	R/D2
Fully correct.	0.77	0.01	0	0.74	0	0
Correct shape and rule. Wrong colour.	0	0.02	0	0	0.01	0
Correct rule. Wrong shape.	0.08	0.12	0.05	0.08	0.53	0.17
Correct shape. Wrong rule.	0.03	0.18	0.02	0.02	0.01	0
All wrong.	0.12	0.67	0.93	0.16	0.46	0.83

TABLE II: Percentage of types of errors for each world model / distance for the OCL architecture.



(a) Pointer network visualisation



(b) Compositional representation measure

Fig. 5: Representation visualisation and compositionality metric. (a) Visualisation of the pointer networks’ pixel copying mechanism. The colour coding (black to blue for the x axis and black to red for the y axis) denotes the input index of a pixel. In the input images the colours are distributed in order. In the output images the colours denote the pixel index of the input image each pixel was copied from. The light green tinge on some pixels in the images show the pixels that were part of the coloured objects in the input. For each test set (world model + distance) we show how each of the two axial pointer networks copies pixels from input to output both in the case of correct prediction (lines of correct) and of a wrong one (lines of error). For each network, world model, distance combination the left image is the input and the right the output image. Where we provide no visualisations (e.g. APLN, Translate, Distance 0, error or APN, Rotate, Distance 2, correct) is because for these cases there were no samples generated. (b) The results of the measure of compositional representation M for all both axial pointer networks (APLN: red bars, APN: green bars) for all world models and distances. The left bar plot shows the object pixel copying error while the right one shows the error in the copying of the rest of the pixels (canvas pixels).

Rule	Correct	Wrong colour	Wrong shape	Wrong rule	All wrong
Left	247	17	137	200	1321
Up	532	0	107	28	408
90	511	11	104	16	398
180	225	0	684	4	1044

TABLE III: Number of types of errors for each rule for the OCL architecture.

C. The Copying Layer of the Axial Pointer Architectures Reveals Non-Compositional Structure

As noted above, the axial pointer architecture enables direct visualisation (Fig. 5(a)) of the pixel-level copying operation in the final layer, through which we can assess whether any network has learned to organise its copying operations in a compositionally structured manner.

The first observation is that the APLN groups copied pixels into rows and columns, as intended by design. Comparing the

APLN’s copying patterns between the Translate and Rotate world models, it is precisely this row–column grouping that enables strong OOD performance on the Translate model while causing failure on the Rotate model. Figs. 5A-iv, A-v, and A-vi show that, on correct predictions in the Rotate model, the APLN occasionally leverages its bias to approximate the rotated output by rearranging multiple rows and columns.

The visualisations for the APN across all world model and distance combinations reveal that this architecture also copies pixels in spatially coherent groups. However, these groups exhibit noisier boundaries and less consistent internal structure compared to the row–column groups of the APLN. Despite having the capacity to form pixel groups corresponding to objects and to learn representations of their transformations as affine operations (translation and rotation), the APN never develops such latent features.

To provide a quantitative assessment of the networks’ failure to form latent variables corresponding to the compositional structure of the tasks, we define a metric of divergence from the compositional representation. This metric measures the average discrepancy between the observed pixel displacement and the displacement expected under a compositionally correct operation that acts solely on the coloured objects. Specifically, the metric M is defined as:

$$M = \left[\frac{\frac{1}{C} \sum_{c=0}^C \|p_c^* - p_c\|^2}{\frac{1}{C+B} \sum_{i=0}^{C+B} \|p_i^{rand} - p_i\|^2}, \frac{\frac{1}{B} \sum_{b=0}^B \|p_b^* - p_b\|^2}{\frac{1}{C+B} \sum_{i=0}^{C+B} \|p_i^{rand} - p_i\|^2} \right]$$

where C denotes the number of coloured (object) pixels and B the number of black (canvas) pixels in the image. The term p_c^* is the expected displacement of object pixel c under compositionally correct behaviour, i.e., the displacement that would result from detecting and transforming the coloured object. The term p_b^* is the expected displacement of canvas pixel b , defined as 0. The terms p_c and p_b are the actual displacements applied by the network to object pixel c and canvas pixel b , respectively. The term p_i^{rand} is the displacement of pixel i under a random permutation of all pixel positions.

The metric decomposes into two components. The first quantifies how accurately the network displaces object pixels according to the task’s prescribed action. The second measures the degree to which the network distinguishes between object and canvas pixels, thereby assessing whether the copying layer has acquired the concept of a coloured shape. Fig. 5(b) shows that in most network/distance combinations, the networks succeed in detecting and correctly displacing the object pixels (with the exception of the APN on the Rotate world model). However, in all cases they fail to discriminate between object and canvas pixels, with most configurations exhibiting approximately 50% divergence from the random baseline for canvas pixel displacement. A notable exception is the APLN at Distance 1 and Distance 2 on the Rotate world model, where the network successfully minimises canvas pixel displacement but, due to its inductive bias, still displaces entire rows or columns (as shown in Figs. 5A-v and A-vi).

In order to validate the metric we run it on the results of a hand crafted function that took the input images and rule denoting language bits as input, discovered the input objects and followed the correct rule to reposition the object, creating

the output image. At the same time it kept track of each pixel displacement (much like the APN and APLN architectures). Running the metric on these displacements we generated the metric’s theoretical minimum for the object and canvas pixels for all world models and distances (see Table IV).

Pixel type	T/D0	T/D1	T/D2	R/D0	R/D1	R/D2
Mean Object pixel displacement.	0	0	0	0.0036	0.0049	0.005
Mean Canvas pixel displacement.	0	0	0	0	0	0

TABLE IV: Object and Canvas mean pixel displacements for the results of a function crafted to produce the correct output.

V. DISCUSSION

Our experiments reveal a fundamental gap between OOD task performance and the acquisition of compositional representations. Both the three baseline networks (MLP, CNN, and Transformer) and the composition-biased OCL model achieve very poor results in the Distance 1 and Distance 2 settings for both world models. While the APLN achieves near-perfect accuracy on the Translate world model, inspection of its learned representations demonstrates that this success is attributable to a hand-crafted inductive bias rather than to the emergence of genuinely compositional latent structure. These findings have direct implications for how the community evaluates claims of compositional generalisation: benchmark accuracy on held-out distribution shifts, even when carefully controlled, does not constitute sufficient evidence that a model has learned reusable, composable primitives.

Several limitations of the present study should be acknowledged. First, we evaluate a relatively small set of architectures. Extending the analysis to a broader range of models—including graph neural networks, state-space models, and neurosymbolic architectures specifically designed for systematic generalisation—would provide a more comprehensive picture of the current landscape. The OCL model used here was the only example of such an object-based, composition-capable architecture, and its failure to generalise OOD highlights the need for more extensive testing of such models. Second, hyperparameter search was limited in scope. Because all models (except OCL) achieve ceiling performance on the in-distribution test set, there is no in-distribution validation signal to drive automated hyperparameter optimisation for OOD performance. We consider it unlikely that alternative hyperparameter configurations would qualitatively alter the relative ordering of the tested architectures on OOD sets, though this remains an open empirical question. For OCL, we kept the hyperparameters practically identical to those used by its creators to demonstrate its compositional capabilities. We believe this yields a fairer comparison than tuning the hyperparameters ourselves to reach ceiling performance on the training set. Third, the current world models employ atomic actions (single translations or rotations). We do not explore settings with compositional action spaces—i.e., tasks where the target transformation is itself a composition of primitive operations—which would enable evaluation of a broader class

of models, including language-conditioned architectures such as that of Lake and Baroni. That said, the OCL example points to the need to distinguish between complex compositional chains over object attributes, where a large number of bespoke networks achieve good results, and OOD generalisation over both object and rule composition, where, we argue, the same architectures may be less appropriate. Keeping the world models compositionally simple but demanding in terms of OOD capability is one way to partially achieve this.

Several directions naturally extend the present framework. A high-priority extension is the introduction of compositional action spaces, where output transformations are sequences or combinations of primitive operations. This would enable direct comparison with meta-learning and in-context learning approaches that rely on linguistic or symbolic task specifications. Additionally, scaling the world models to higher-dimensional observation spaces and richer object ontologies would test whether the dissociation between performance and compositional representation persists under more complex distributional shifts.

On the evaluation side, the development of automated, architecture-agnostic probing methods for compositional structure in latent spaces—beyond the pointer-network visualisation employed here—would make the proposed evaluation paradigm applicable to arbitrary deep networks. Techniques from mechanistic interpretability, such as activation patching and causal tracing, offer promising tools for this purpose.

Finally, it would be valuable to investigate whether training objectives that explicitly incentivise compressed, factored representations—such as information-bottleneck regularisation or disentanglement penalties—can bridge the gap between OOD performance and genuine compositionality that our results expose. Connecting these findings to the complexity-theoretic framework of Elmoznino et al. through empirical measurement of representational Kolmogorov complexity would help ground future compositionality claims in more rigorous theoretical criteria.

VI. ACKNOWLEDGMENTS

GD and SS acknowledge the support of the Engineering and Physical Sciences Research Council (EPSRC) grant “UKRI1076 Element-compositor methods for out-of-distribution machine learning”.

For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

We used Claude Opus 4.6 Extended as an editor. The whole manuscript was written manually by the authors and then given to Claude to do small changes in the format of phrases and the choice of certain words. The proposed changes were then re-reviewed by the authors.

REFERENCES

- [1] K. M. Collins, I. Sucholutsky, U. Bhatt, K. Chandra, L. Wong, M. Lee, C. E. Zhang, T. Zhi-Xuan, M. Ho, V. Mansinghka, A. Weller, J. B. Tenenbaum, and T. L. Griffiths, “Building machines that learn and think with people,” *Nature Human Behaviour*, vol. 8, no. 10, pp. 1851–1863, Oct. 2024.
- [2] A. L. Foote and J. D. Crystal, “Metacognition in the rat,” *Current biology: CB*, vol. 17, no. 6, pp. 551–555, Mar. 2007.
- [3] M. Lázaro-Gredilla, D. Lin, J. S. Guntupalli, and D. George, “Beyond imitation: Zero-shot task transfer on robots by learning concepts as cognitive programs,” *Science Robotics*, vol. 4, no. 26, p. eaav3150, Jan. 2019.
- [4] A. Gopnik, C. Glymour, D. M. Sobel, L. E. Schulz, T. Kushnir, and D. Danks, “A Theory of Causal Learning in Children: Causal Maps and Bayes Nets,” *Psychological Review*, vol. 111, no. 1, pp. 3–32, 2004.
- [5] L. Luettgau, T. Erdmann, S. Veselic, K. L. Stachenfeld, Z. Kurth-Nelson, R. Moran, and R. J. Dolan, “Decomposing dynamical subprocesses for compositional generalization,” *Proceedings of the National Academy of Sciences*, vol. 121, no. 46, p. e2408134121, Nov. 2024.
- [6] Z. G. Szabó, “The case for compositionality,” in *The Oxford Handbook of Compositionality*, M. Werning, W. Hinzen, and E. Machery, Eds. Oxford: Oxford University Press, Feb. 2012, pp. 64–80.
- [7] N. Gontier, “Combinatoriality and Compositionality in Everyday Primate Skills,” *International Journal of Primatology*, vol. 45, no. 3, pp. 563–588, Jun. 2024.
- [8] S. Dehaene, F. Al Roumi, Y. Lakretz, S. Planton, and M. Sable-Meyer, “Symbols and mental programs: A hypothesis about human singularity,” *Trends in Cognitive Sciences*, vol. 26, no. 9, pp. 751–766, Sep. 2022.
- [9] A. Goyal and Y. Bengio, “Inductive biases for deep learning of higher-level cognition,” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 478, no. 2266, p. 20210068, Oct. 2022.
- [10] J. Russin, S. W. McGrath, D. J. Williams, and L. Elber-Dorozko, “From Frege to chatGPT: Compositionality in language, cognition, and deep neural networks,” May 2024.
- [11] A. d’Avila Garcez and L. C. Lamb, “Neurosymbolic AI: The 3rd wave,” *Artificial Intelligence Review*, vol. 56, no. 11, pp. 12 387–12 406, Nov. 2023.
- [12] K. Ellis, C. Wong, M. Nye, M. Sablé-Meyer, L. Morales, L. Hewitt, L. Cary, A. Solar-Lezama, and J. B. Tenenbaum, “DreamCoder: Bootstrapping inductive program synthesis with wake-sleep library learning,” in *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, ser. PLDI 2021. New York, NY, USA: Association for Computing Machinery, Jun. 2021, pp. 835–850.
- [13] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, “Beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework,” in *International Conference on Learning Representations*, Feb. 2017.
- [14] Y.-F. Wu, M. Lee, and S. Ahn, “Neural Language of Thought Models,” in *The Twelfth International Conference on Learning Representations*, 2024.
- [15] T. Kipf, E. van der Pol, and M. Welling, “Contrastive Learning of Structured World Models,” in *Eighth International Conference on Learning Representations*, Apr. 2020.
- [16] S. Venkatraman, M. Jain, L. Scimeca, M. Kim, M. Sendera, M. Hasan, L. Rowe, S. Mittal, P. Lemos, E. Bengio, A. Adam, J. Rector-Brooks, Y. Bengio, G. Berseth, and N. Malkin, “Amortizing intractable inference in diffusion models for vision, language, and control,” in *Advances in Neural Information Processing Systems*, vol. 37. Curran Associates, Inc., 2024.
- [17] D. Hupkes, V. Dankers, M. Mul, and E. Bruni, “Compositionality Decomposed: How do Neural Networks Generalise?” *Journal of Artificial Intelligence Research*, vol. 67, pp. 757–795, Apr. 2020.
- [18] J. A. Fodor and Z. W. Pylyshyn, “Connectionism and cognitive architecture: A critical analysis,” *Cognition*, vol. 28, no. 1, pp. 3–71, Mar. 1988.
- [19] T. Van Gelder, “Compositionality: A connectionist variation on a classical theme,” *Cognitive Science*, vol. 14, no. 3, pp. 355–384, Jul. 1990.
- [20] S. Thorpe, “Local vs. Distributed Coding,” *Intellectica*, vol. 8, no. 2, pp. 3–40, 1989.
- [21] E. Elmoznino, T. Jiralerspong, Y. Bengio, and G. Lajoie, “A Complexity-Based Theory of Compositionality,” Feb. 2025.
- [22] B. M. Lake and M. Baroni, “Human-like systematic generalization through a meta-learning neural network,” *Nature*, vol. 623, no. 7985, pp. 115–121, Nov. 2023.
- [23] F. Chollet, “On the Measure of Intelligence,” Nov. 2019.
- [24] —, “Fchollet/ARC-AGI: The Abstraction and Reasoning Corpus,” GitHub repository, Apr. 2025. [Online]. Available: <https://github.com/fchollet/ARC-AGI>
- [25] A. Zerroug, M. Vaishnav, J. Colin, S. Musslick, and T. Serre, “A Benchmark for Compositional Visual Reasoning,” in *Advances in Neural*

- Information Processing Systems*, vol. 35. Curran Associates, Inc., 2022, pp. 29 776–29 788.
- [26] S. Depeweg, C. A. Rothkopf, and F. Jäkel, “Solving Bongard Problems With a Visual Language and Pragmatic Constraints,” *Cognitive Science*, vol. 48, no. 5, p. e13432, 2024.
 - [27] I. Dasgupta, D. Guo, S. J. Gershman, and N. D. Goodman, “Analyzing Machine-Learned Representations: A Natural Language Case Study,” *Cognitive Science*, vol. 44, no. 12, p. e12925, Dec. 2020.
 - [28] M. Okawa, E. S. Lubana, R. P. Dick, and H. Tanaka, “Compositional abilities emerge multiplicatively: Exploring diffusion models on a synthetic task,” in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS ’23. Red Hook, NY, USA: Curran Associates Inc., Dec. 2023, pp. 50 173–50 195.
 - [29] B. Dedhia, M. Chang, J. C. Snell, T. L. Griffiths, and N. K. Jha, “Impromptu: In-context composition from image prompts,” in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS ’23. Red Hook, NY, USA: Curran Associates Inc., Dec. 2023, pp. 53 261–53 274.
 - [30] Z. G. Szabó, “Compositionality,” in *The Stanford Encyclopedia of Philosophy*, summer 2004 ed., E. N. Zalta, Ed. Metaphysics Research Lab, Stanford University, 2004. [Online]. Available: <https://plato.stanford.edu/archives/sum2004/entries/compositionality/>
 - [31] M. Werning, W. Hinzen, and E. Machery, Eds., *The Oxford Handbook of Compositionality*. Oxford: Oxford University Press, Feb. 2012.
 - [32] J. A. Fodor, *The Language of Thought*. Cambridge, Mass: Harvard University Press, Jan. 1980.
 - [33] B. H. Partee, “Lexical Semantics and Compositionality,” in *An Invitation to Cognitive Science, Volume 1: Language*, 2nd ed., L. R. Gleitman and M. Liberman, Eds. Cambridge, MA: The MIT Press, 1995, pp. 311–360.
 - [34] T. R. Zentall, E. A. Wasserman, O. F. Lazareva, R. K. R. Thompson, and M. J. Rattermann, “Concept Learning in Animals,” *Comparative Cognition & Behavior Reviews*, vol. 3, 2008.
 - [35] F. P. Battaglia, G. Borensztajn, and R. Bod, “Structured cognition and neural systems: From rats to language,” *Neuroscience & Biobehavioral Reviews*, vol. 36, no. 7, pp. 1626–1639, Aug. 2012.
 - [36] S. Ohlsson and E. Lehtinen, “Abstraction and the acquisition of complex ideas,” *International Journal of Educational Research*, vol. 27, no. 1, pp. 37–48, Jan. 1997.
 - [37] J. B. Tenenbaum, C. Kemp, T. L. Griffiths, and N. D. Goodman, “How to Grow a Mind: Statistics, Structure, and Abstraction,” *Science*, vol. 331, no. 6022, pp. 1279–1285, Mar. 2011.
 - [38] B. Lin, D. Bouneffouf, and I. Rish, “A Survey on Compositional Generalization in Applications,” Feb. 2023.
 - [39] S. Sinha, T. Premisri, and P. Kordjamshidi, “A Survey on Compositional Learning of AI Models: Theoretical and Experimental Practices,” *Transactions on Machine Learning Research*, 2024. [Online]. Available: <https://openreview.net/forum?id=BXDxwItNqQ>
 - [40] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
 - [41] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, “Handwritten digit recognition with a back-propagation network,” in *Advances in Neural Information Processing Systems*, D. Touretzky, Ed., vol. 2. Morgan-Kaufmann, 1989, pp. 396–404.
 - [42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention Is All You Need,” in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017, pp. 5998–6008.
 - [43] G. Singh, F. Deng, and S. Ahn, “Illiterate DALL-E Learns to Compose,” in *International Conference on Learning Representations*, 2022.
 - [44] C. Olah, “Distributed Representations: Composition & Superposition,” <https://transformer-circuits.pub/2023/superposition-composition/index.html>, May 2023, transformer Circuits Thread. [Online]. Available: <https://transformer-circuits.pub/2023/superposition-composition/index.html>
 - [45] H. Liu, Z. Dai, D. So, and Q. V. Le, “Pay Attention to MLPs,” in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 9204–9215.
 - [46] J. Ho, N. Kalchbrenner, D. Weissenborn, and T. Salimans, “Axial Attention in Multidimensional Transformers,” Dec. 2019.
 - [47] O. Vinyals, M. Fortunato, and N. Jaitly, “Pointer networks,” in *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS’15, vol. 2. Cambridge, MA, USA: MIT Press, Dec. 2015, pp. 2692–2700.

George Dimitriadis obtained a B.S. in Biology from the Aristoteles University of Thessaloniki, Greece, a B.S. in Physics and a B.S. in Psychology from the Open University, UK and a Ph.D. in Protein Engineering from the Department of Physics and Astronomy of the University of Leeds, UK. He was a Research Associate in Experimental Neuroscience at the Donders Centre for Cognition in Radboud University, the Netherlands and a Senior Research Associate at the Sainsbury Wellcome Centre (SWC), UCL, UK. He was a Joint Senior Research Fellow in Experimental and Theoretical Neuroscience in SWC and the Gatsby Computational Neuroscience Unit, UCL, UK. He is currently a Research Associate in the School of Electronic Engineering and Computer Science in the University of Essex, UK.

Spyridon Samothrakis received the Ph.D. degree in computer science in 2014, with a focus on optimal game playing in multi-player games. He is currently a Senior Chief Scientific Adviser and Senior Lecturer at the School of Computer Science and Electronic Engineering (CSEE), University of Essex. His current research interests include meta-learning and reinforcement learning.