Acquiring Complex Concepts with Comparative Learning

Diego Calanzone[1] Filippo Merlo[2]
[1] DISI, University of Trento
[2] CIMEC, University of Trento
{diego.calanzone, filippo.merlo}@studenti.unitn.it

April 10, 2024

Abstract

Current vision and language models (VLMs) can acquire simple and complex notions in the form of linguistic expressions grounded on visual inputs. Traditional training setups for these involve noisy examples from internet-scraped data that can lead to inefficient and coarse learning. Taking from literature in cognitive science and developmental psychology, learning frameworks employed by humans could be considered as new training paradigms for VLMs: for instance, learning gradually more complex notions (Progressive Alignment) by comparing examples in a controlled environment (Comparative Learning) could nudge compositionality as ability. Current approaches either consist of training monolithic networks on sparse concepts of varying complexity, or task-specific modules are specialized with narrow knowledge. In this study we advance two questions: can we train a single, multi-task network to learn primitive concepts? Moreover, can we leverage Comparative Learning to acquire more complex notions, such as the logical composition of such primitives?

1 Introduction

1.1 Human-Inspired Progressive Alignment

How does an agent learn concepts? In deep learning literature, an emerging candidate answer is "by contrast": a model is taught affinities in training data by clustering between positive and negative samples, with none or minimal amounts of human-crafted labels. This approach, namely *Contrastive Learning (CL)*, links with theories in developmental psychology from Gentner et al. [9, 10]: with comparison, children can find commonalities and differences in input elements; moreover, they can also understand potential structures. Selecting relevant features in the inputs is another crucial skill: in Contrastive Learning, the peculiarity of details in a learned concept strongly depends on how the negative examples differ from the positive ones [23].

Learning by comparison and focusing on relevant information are key features of a novel learning framework proposed by Bao et al. [3]: an agent learns one single-word concept at a time by seeing objects that share such notion as common property, contrarily to objects that do not. For instance, let the learned attribute be e.g. “red”: positive samples would consist of images sharing this feature, e.g. "red plastic cone" and "red metallic sphere", with respect

to negative samples that may share multiple features but exclude the target, e.g. "brown plastic cone" and "blue plastic cone". This controlled learning setup allows to truly focus on a target concept, as the comparison task is designed to highlight it in order to successfully distinguish objects; consequently, concept-specific representations can be concatenated to describe composite object properties, e.g. "red, spherical, metallic object". This framework is named "Comparative Learning with Progressive Alignment" [3] and it's shown to be effective in recognizing multiple features of an object and in imagining new ones; this approach contributes to the objective of instilling compositionality capabilities and hierarchies of concepts in artificial neural networks.

1.2 Scalable multi-concept networks

Being able to distinguish examples for different concepts can be seen as a set of classification tasks for multiple distributions or domains: this problem is tackled in deep learning with methods aiming for generality (e.g. domain adaptation [8], multi-task learning [7], modular deep learning [18]) and consistent performance with increasing knowledge over time (i.e. continual learning [22]).

In context of progressive-alignment theory, Bao et al. [3] introduce a modular memory of independent neural networks, each dedicated to a learned notion. This setup is not vulnerable to catastrophic forgetting, a main challenge in continual learning, as further training knowledge is not compressed in a single, parametric model and the memory of concepts can keep expanding according to the given computational resources. This methodology, however, introduces a high level of redundancy as specific neural components are isolated in a discrete space (the memory), with no possibility to leverage similarities for efficient sampling or to perform constructive operations in the conceptual space without handcrafted techniques. Contrarily, generative models such as Denoising Diffusion Probabilistic Models [12] encode concepts in a single continuous space, allowing for efficient information compression and interpolation.

Consequently, our first objective is to extend progressive alignment to efficient generative models. We propose a unified architecture for complex concept learning based on two levels of abstraction: given a conceptual description in natural language, a hyper-network, that is a neural network that generates weights for another neural network [5], predicts parameters for a second one, which encodes and allocates positive and negative instances of a notion in a dedicated geometrical space. Memory models based on dense representational spaces are known to be efficient and scalable [12, 13], we expand on this approach by robustly learning concepts in a fine-grained procedure with Comparative Learning.

1.3 Beyond arithmetics: logical concepts

Another limitation we identified in the study from Bao et al. [3] is the acquisition of single-word notions: this voluntary simplification does not however represent reality, in which complex natural language expressions are used, e.g. "four-legged and hairy mammal". Moreover, composing notions may require a higher level of abstraction beyond simple concatenation, that is reasoning on logical expressions and thus more general sets of positive and negative examples.

We aim to teach the model logical relations between simple concepts through the same learning procedure proposed by Bao et al [3]. In the original study, the authors trained a model on single attributes by having it comparing positive examples with negative ones, with the only distinction being the target notion. In our approach, we expand on this method by introducing sets where positive examples adhere to the specified logical relation, and negative examples

deviate from it. For instance, to learn the "and" logical relation we prepare positive examples that describe the joint features, eg. "red and cone" or "sphere and metallic", and negatives that violate the rule, eg. "red and spherical" or "sphere and plastic". In the spirit of Progressive Alignment [11], with this framework, we aim to teach artificial networks increasingly more abstract concepts and relations.

2 Related work

Visual-Language models (VLMs) can link complex linguistic expressions to images [20] [15], but is this induced by reasoning on underlying simple concepts? Pavlick et al. [25] expand on this question by analyzing a set of state-of-the-art VLMs: given a universal vocabulary of base notions, e.g. "red" or "wing", they design descriptive prompts to query a given image; the words with a high matching score, e.g. cosine similarity, for the given image define the set of primitive features. Since each input image originally comes with a complex linguistic expression, e.g. "a red-winged black bird", the goal is to find a relation between the primitives queried with VLMs and the given caption. With the assumption that embeddings can be composed arithmetically, a vision widely shared in literature [17], a linear classifier is trained to map the linear combination of primitives to a caption embedding; consequently, given an input image the extracted set of basic notions should link to the right caption. The experiments reported by Pavlick et al. [25] show weak correlation scores: particularly, spurious relationships, e.g. attributes describing the wrong entities, such as the background, can drive the classifier to predict the wrong caption. This may suggest that current VLMs ground simple and complex linguistic expressions separately, without exploiting compositional properties; another hypothesis is that the level of understanding of textual inputs is coarse.

Building on this intuition, Yuksekgonul et al. [24] test whether VLMs act as bag of words models, that is whether they link linguistic expressions to visual inputs only because of the presence of keywords and regardless of the syntactic structure of the caption. VLMs are shown to be insensitive to swapping references in text and thus to changes in their semantic meaning: this further suggests that fine-grained relationships of attributes are not acquired. Conversely, augmenting contrastive examples with such textual variations and meaning shifts improves performance in fine-grained grounding tasks. In connection to what is thus advanced by Pavlick et al. [25]: primitive concepts can show high affinity with visual inputs regardless of the structure of the provided caption, suggesting that current VLMs may not organize concepts in hierarchical structures induced by composition.

Moreover, complex notions may not only consist of simple arithmetics: logical expressions are used as queries for visual inputs, e.g. "an object that is red or green and it's either spherical or a cube". Visual reasoning is a hard task that is tackled with neuro-symbolic approaches [14] [2] or articulated frameworks for language models [1] [6].

From a different perspective, we frame this as a multi-task learning problem: primitive concepts are basic skills that could be composed in a non-trivial way to more ones. Research in Modular Deep Learning [18] studies a variety of approaches to building multi-task learners: hyper-networks are recently gaining traction, as they allow to generate neural networks given the task and further specifics [16]; alternatively, more parameter-efficient approaches [19] aim at learning a fixed set of n_s skills (task-specific networks) that are linearly combined in order to solve a larger set of tasks n_t, $n_t >> n_s$. Within this research line, we frame the work from Bao et al. [3] as similar to Progressive Neural Networks [21]: the set of task-specific neural

components is expanded as new notions are explicitly presented.

3 Proposed Methodology

3.1 Dataset

We train and evaluate our models on the SOLA dataset (Simulated Objects for Language Acquisition), created by Bao et al. [3]. SOLA, characterized by low noise and distinct attributes, facilitates efficient sample comparisons and mapping of language features to grounded concepts. SOLA comprises images of simulated objects, each associated with three learning attributes: color, material, and shape.

These objects present a combination of 8 colors, 11 shapes, and 4 materials, with additional diversity introduced through variations in light settings (3 settings per object) and camera angles (6 angles per object). In total, it contains 6336 Red Green Blue Alpha (RGBA) images of synthetic objects.

To assess model generalizability and robustness to noisy inputs, a Variation Test set (D_{test_v}) was created, consisting of 989 RGBA images subjected to transformations such as stretching, shade changes, or size alterations. Objects in this test set underwent modifications along the x, y, and z axes, changes in shade darkness or lightness, and resizing to medium or small dimensions.

The dataset was split into various sets, as outlined in Table 1. A Novel Composition Test set (D_{test_nc}) was reserved to evaluate the novel composition capability of the methods, with 9 exclusive learning attribute pairs, while the remaining pairs were incorporated into the Train set (D_{train}) for word acquisition training.

For complex concept acquisition, we generate a new training dataset ($D_{train\_complex}$) by merging the D_{train} and D_{test_nc} datasets (comprising all non-augmented objects). Subsequently, we subtract a set of 32 objects from this combined dataset to create a test dataset D_{test_no}, composed of 576 objects unseen during training. This approach is essential as complex concepts involve the composition of two simple concepts and we need to ensure that the model encounters all possible pairs during training. The composition of a test set of unseen objects allows the evaluation of the model's proficiency in recognizing acquired complex concepts in novel scenarios.

Table 1: Splits on RGBA Images

Split	Num Objects
D_{train}	5094
$D_{train\_complex}$	5760
D_{test_nc}	1242
D_{test_no}	576
D_{test_v}	989

3.2 Comparative Learning

Comparative learning is defined by Bao et al. [3] as "the process of finding the similarities and differences from a set of inputs". The general formulation states that for each learned concept l_i in an unconstrained set $L = \{l_1, l_2, \ldots\}$, must be assembled a batch of samples

$\mathscr{B}_s = \{a_1^{l_i}, \ldots, a_n^{l_i}\}$, that share the label l_i for similarity learning, and a batch of samples $\mathscr{B}_d = \{b_1^{l_j}, \ldots, b_n^{l_j}\}, j \neq i$ that cannot be described by l_i for difference learning. The process of SIM_{l_i} (1) finds the similarities among the examples in $\mathscr{B}_s$, and extract out the representation REP_{l_i} expected to refer to l_i. The process of DIFF_{l_i} (2) highlights the differences between l_i and other non-compatible labels refining the representation REP_{l_i}.

$$\text{REP}_{l_i} = \text{SIM}_{l_i}(\{a^{l_i} \in \mathscr{B}_s\}) \tag{1}$$

$$\text{REP}_{l_i} = \text{DIFF}_{l_i}(a^{l_i}, \{b^{l_j} \in \mathscr{B}_d\}) \tag{2}$$

3.3 A baseline for primitive concepts

Bao et al. [3] contextualize this training method with a set of fixed visual inputs. For each concept, e.g., "red," they assemble a batch of images sharing it for similarity training and a batch of images that are of any other color (non-compatible) but "red" for difference refinement. They keep the rest of the attributes the same for better structural alignment.

In their setting, each image of the batches a_u goes through a pre-trained frozen CLIP image embedding as the first step. Subsequently, all the image embeddings e_u undergo two processes. The former is information denoising, where the elementwise product of each image embedding and a filter F_{l_i} is computed. The filter is a learning vector the same size as the embedding that masks the input embedding by selecting only the relevant dimensions that contribute to the learned concept l_i. This masked embedding then goes through the attention establishment process, passing two fully connected layers of an encoder Enc_{l_i}, to output a condensed representation r_u.

On top of learning the attention filtration process (F_{l_i}, Enc_{l_i}), the centroid is computed for all the sample representations r_u from the similarity batch $\mathscr{B}_s$ as the condensed representation REP_{l_i} for l_i. For difference learning, all the $\mathscr{B}_d$ samples go through the same filtration and encoding process for the concept l_i. Finally, the loss function pushes the distance between each sim batch sample and the centroid close, and it further pushes the diff batch sample representations and the centroid apart.

This approach jointly trains the filter, the encoder, and the representation, producing a different set of these three objects for each of the learned concepts $\{l_i : [F_{l_i}, Enc_{l_i}, \text{REP}_{l_i}]\}$. Our first goal is to unify the learning of multiple concepts under the same single architecture while keeping the same training process.

3.4 Hyper-networks for primitive concepts

To overcome the increasing computational cost of a discrete memory of neural networks, we frame the problem of acquiring multiple concepts as multitask learning: distinguishing objects by the notion that they represent is a classification task. We thus employ a Hyper-Network [5], namely $\mathscr{H}_\theta$, that is a neural network that given a task representation $\tau_{l_i} \in \mathbb{R}^h$, it predicts a set of weights for a task-specific network $\mathscr{N}_\phi$, such that:

$$\phi_{l_i} = \mathscr{H}_\theta(\tau_{l_i}), \;\; l_i \in L \tag{3}$$

For $\mathscr{N}_\phi$ we adopt the same architecture proposed by Bao et al. [3], consisting in the filter, the encoder and the *learnable* prototypical representation (or "centroid"). We exclude the decoder component from our framework as we focus on concept learning and we use different techniques for composition and reasoning.

As in the original work, the encoder consists of a downsampling linear layer that compresses the input features to a lower-dimensional space, followed by an upsampling projection to the output dimensionality. The decoder consists of four upsampling linear layers as originally defined, while the filter and the centroid are single predicted vectors.
We take inspiration from HyperFormer, Mahbadi et al. [16]: our hyper-network $\mathscr{H}_\theta$ consists in a multi-layer perceptron $\mathbf{h}^j$ for each linear layer j, to predict task-specific weights ϕ_j.

$$\begin{aligned} \tau_{l_i} = \texttt{CLIP}_t(l_i),\ \ \mathscr{H}_\theta = \{\mathbf{h}^{\text{F}}, \mathbf{h}^{\text{Enc}}, \mathbf{h}^{\text{REP}}\} \\ \mathscr{N}_{\phi_{l_i}} = \{F_{l_i}, Enc_{l_i}, REP_{l_i}\},\ \ \phi_{l_i} = \{\phi_{l_i}^{\text{F}},\ \phi_{l_i}^{\text{Enc}},\ \phi_{l_i}^{\text{REP}}\} \\ \phi_{l_i}^{\text{F}} = \mathbf{h}^{\text{F}}(l_i) \\ \phi_{l_i}^{\text{Enc}} = \mathbf{h}^{\text{Enc}}(l_i) \\ \phi_{l_i}^{\text{REP}} = \mathbf{h}^{\text{REP}}(l_i) \end{aligned} \tag{4}$$

Each hyper-MLP has the same architecture: one linear layer predicts the projection weight matrix W and a second one predicts the bias b.

$$\phi^k = \mathbf{h}^k(l_i) = \{W_k, b_k\} \tag{5}$$

The learnable REP_{l_i} has been introduced as a replacement for computing the prototypical concept embedding from the average of positive samples, as originally proposed by Bao et al. [3]: this allows the network to directly predict the prototypical representation given the task prompt and we show in our experiments that it further stabilizes training.

As concept lessons are represented by linguistic expressions, we employ CLIP's textual encoder to embed a notion string into a task embedding $\tau_{l_i} \in \mathbb{R}^h$, in this case, $h = 512$.

3.4.1 Casting catastrophic forgetting

Multitask learning architectures such as hyper-networks are subject to the problem of forgetting examples seen early at training time while optimizing for new ones: this is due to the fact that firstly the incoming samples may be part of a novel data distribution and thus the network will gradually shift to new centroids and latent spaces.

To remember past examples, either an additional loss can be added to keep the old concept representations fixed (a regularizer), or past data points could be re-proposed to the network with a "replay buffer".

We employ a recent, successful technique that combines these two approaches: Dark Experience Replay (DER++), introduced by Buzzega et al. [4]. A replay buffer can ensure uniformity in the stored sample classes with a sampling technique named reservoir strategy: random data points, e.g. visual examples, are remembered (in rotation) in the buffer, along with the hidden representations computed by our network at the current time.
Consequently, our hyper-network $\mathscr{H}_\theta$ will optimize against the original losses for comparative learning, for each concept $l_i \in L$:

$$\begin{aligned} \mathscr{L}_{SIM_{l_i}} &= \sum_{u}^{|\mathscr{B}_s|} \texttt{Dist}(r_u^{l_i}, REP_{l_i}) \\ \mathscr{L}_{DIF_{l_i}} &= \sum_{v}^{|\mathscr{B}_d|} \texttt{Dist}(r_v^{l_i}, REP_{l_i}) \\ \mathscr{L}_{\text{COMP}_{l_i}} &= (\mathscr{L}_{SIM_{l_i}})^2 + (1 - \mathscr{L}_{DIF_{l_i}})^2 \end{aligned} \tag{6}$$

Given a vector distance function `Dist`, such as mean squared error.

Furthermore, two additional objectives are added to tackle catastrophic forgetting. As introduced in DER++, we draw two random samples for two arbitrary concepts, from the replay buffer, namely $s_1^{l_f} = (a_1^{l_f}, r_1^{l_f})$ and $s_2^{l_g} = (a_2^{l_g}, r_2^{l_g})$, that is the image sample a and the hidden representation r: s_1 is used to compute the loss $\mathscr{L}_{\text{logits}_{l_f}}$, which acts as regularizer to enforce the same learned representations over time; s_2 is used to compute the task-specific loss, in this case $\mathscr{L}_{\text{COMP}_{l_g}}$.

$$\begin{aligned} \mathscr{L}_{\text{logits}_{l_f}} &= \texttt{Dist}(Enc_{l_f}(a_1^{l_f}),\, r_1^{l_f})^2 \\ \mathscr{L}_{\text{label}_{l_g}} &= \mathscr{L}_{\text{COMP}_{l_g}} \end{aligned} \tag{7}$$

The final regularized objective thus follows:

$$\mathscr{L} = \mathscr{L}_{\text{COMP}_{l_i}} + \alpha \cdot \mathscr{L}_{\text{logits}} + \beta \cdot \mathscr{L}_{\text{label}} \tag{8}$$

Where α and β are training hyperparameters. This approach has been shown to significantly improve accuracy scores as gradually more concepts are learned by our hyper-network, suffering minimal forgetting with a replay buffer.

3.5 Compound Logical Concepts

Our second research question is whether it is possible to teach more complex concepts through the same training process, beyond primitives learned in the previous experiments such as colors, materials, and shapes. We compose these into logical expressions with basic logic operators: NOT, AND, and OR.

Given an unconstrained set of base concepts $L = \{l_1, l_2, \ldots\}$, we considered all possible logical pairs. For instance, starting from the two simple concepts red, cone, we obtained the set of complex concepts NOT red, NOT cone, red AND cone, red OR cone.

For each complex concept, we created a similarity batch of images with positive samples where the logical relation between the two simple concepts is respected. Additionally, a difference batch with negative samples was generated where the relation is violated. The samples were paired so that, except for the attributes significant for the truth value of the relation, all other features were kept constant.

E.g. for the learning of the concept red AND cone, the similarity batch contained positive images of red cones, while the difference batch included three types of negative samples corresponding to the three cases in which the AND relation is violated:

$$\begin{aligned} \mathscr{B}_s &= \{a \,|\, \text{red AND cone}\} \\ \mathscr{B}_d &= \{a \,|\, \text{red AND NOT cone} \oplus \text{NOT red AND cone} \oplus \text{NOT red AND NOT cone}\} \end{aligned} \tag{9}$$

On the other hand, for the learning of concepts under the OR relation such as metallic OR cube, we generate a similarity batch containing the three different cases where the relation holds and a difference batch where the relation is false:

$$\begin{aligned} \mathscr{B}_s &= \{a \,|\, \text{metallic AND NOT cube} \oplus \text{NOT metallic AND cube} \oplus \text{metallic AND cube}\} \\ \mathscr{B}_d &= \{a \,|\, \text{NOT metallic AND NOT cube}\} \end{aligned} \tag{10}$$

The NOT relation concepts simply present the opposite batch configuration of the simple concepts. The similarity batch contains positive samples where the concept is not present whereas the difference batch has negative samples where it is.

Figure 1: Here is shown an image of a purple plastic torus and the top 10 concepts retrieved during LPR: 1. Aqua or Torus, 2. Red or Torus, 3. Purple or Glass, 4. Brown or Torus, 5. Purple and Plastic, 6. Plastic and Torus, 7. Not Plastic, 8. Purple or Gear, 9. Purple or Rubber, 10. Purple and Torus

To test if comparative learning could be used to learn this kind of complex representation we adopted the model architecture presented by Bao et al. [3].

The model is then evaluated with a modified version of the Multi-Attribute Recognition task presented in [3]. We further tested our more complex hyper-network in this setup: early experiments highlight complications in the training process that we will discuss in the subsequent sections.

4 Evaluation

To test the acquisition of primitives, we employ the same cognitive task introduced by Bao et al. [3]: Multi-Attribute Recognition (MAR). We thus compare the memory-of-networks model (*Baseline*), introduced by the authors, with our multi-task hyper-network (HyperMem).

For complex logical expressions, we modify MAR and thus define Logical Pattern Recognition (LPR). As specified before the only significant change between simple and complex concepts acquisition training is the structure of the similarity and difference batches. In this setup, we train the *Baseline* model only with all the possible logical pairs of primitives.

4.1 Multi-Attribute Recognition

In this task, the models are required to detect all the possible attributes (color, shape, material) associated with a given test image a under two evaluation scenarios: (1) the Novel Composition Setting, where the images have a combination of attributes not encountered during training, i.e., $a \in D_{\text{test\_nc}}$; and (2) the Noisy Setting, where the images in the test set are intentionally subjected to noise, i.e., $a \in D_{\text{test\_v}}$. The models were trained on the training dataset D_{train}.

With the baseline model, for each test image $a_u^{l_k}$ embedded with CLIP, we go through the memory and apply the filter and encoder corresponding to each acquired concept l_k: this yields

the concept-specific hidden representation $r_u^{l_k}$, which indicates how much of that concept is encoded in the test sample. Consequently, for the query image, the presence of all the possible primitives is quantified by a distance function, such as mean squared error (MSE), from the prototypical representation of each notion (the centroid, or REP_{l_k}). We take the top 3 notions for which the encoded input image is the closest to the centroid, which does not necessarily mean that these are respectively a color, a material and a shape; eventually, we compare the top 3 retrieved notions to the ground truth features for that image.

Similarly, we apply the same procedure to our hyper-network, with the only difference that there is no discrete memory to iterate and each linguistic expression describing the primitive is provided as input to the network.

4.1.1 Results

Table 2: Top 3 Accuracy Scores for Multi-Attribute Recognition by concept type (higher is better).

Split	Model	Color	Material	Shape
D_{test_v}	Baseline	0.95	0.75	0.89
	HyperMem	0.56	0.26	0.66
	HyperMem (DER++)	0.74	0.37	0.70
D_{test_nc}	Baseline	0.96	0.48	0.98
	HyperMem	0.37	0.25	0.73
	HyperMem (DER++)	0.71	0.28	0.89

Compatible to the original work, in Table 2 we reproduce high accuracy scores for the baseline model (*baseline*) over distorted visual queries in D_{test_v}: the lowest score is obtained in detecting materials, while the highest corresponds to colors as originally observed [3].

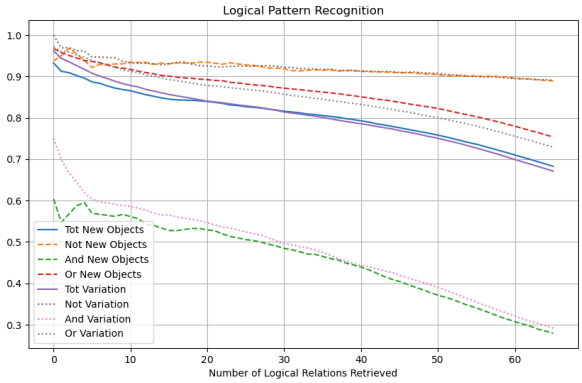
We obtain the same pattern with our hyper-networks (*HyperMem*, *HyperMem (DER++)*), but the overall performance is inferior: task-specific neural network weights are encoded in the hyper-network's continuous latent space, potentially introducing conflict while optimizing for each concept and leading to catastrophic forgetting; this is further supported from the largely inferior performance of our model with no replay buffer (*HyperMem*).

Moreover, the hyper-network architecture determines its multi-task capacity: as in HyperFormer [16], only a single projection layer and a non-linearity function are used to predict another layer's weights.

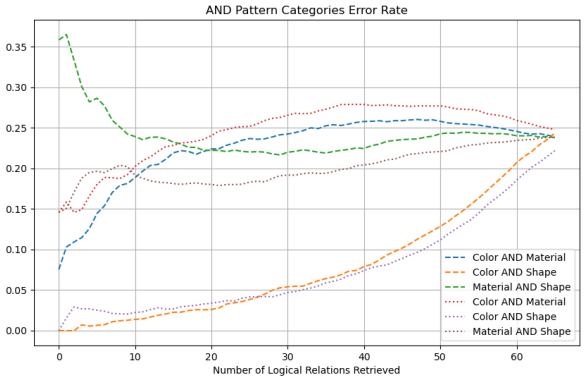
Under the novel composition setting D_{test_nc}, we further observe that our hyper-network augmented with a replay buffer, *HyperMem (DER++)*, achieves comparable but inferior performance with respect to the baseline, despite a smaller gap. This suggests that both models can recognize the presence of basic visual features in objects that differ from any example seen at training time in terms of visual attributes.

4.2 Logical Pattern Recognition

This second evaluation is performed under two different scenarios: (1) the Novel Objects Setting where the test set contains Objects not seen during training and thus presents novel configurations of their three attributes, i.e., $a \in D_{\text{test\_no}}$; and (2) the Noisy Setting, which is the



(a) Logical Pattern Recognition task with varying top-k parameters, revealing higher model scores at lower k values. The model performs well on a subset of valid relations even if not all are precisely matched. AND patterns consistently have lower scores than OR and NOT patterns.



(b) Analysis of low scores on the three AND patterns categories: Color AND Material, Color AND Shape, and Material AND Shape. While Color AND Shape initially have a negligible error rate, it increases as more representations are retrieved. Early trials show elevated error rates in Color AND Material and Material AND Shape, suggesting their contribution to low scores.

Figure 2: Results of the Logical Pattern Recognition evaluation on the model architecture proposed by Bao et al. [3]

same of the previous task. The model was trained on the training dataset for complex concepts acquisition $D_{\text{train\_complex}}$.

To assess the acquisition of complex concepts, we modified the Multi-Attribute Recognition task to align with the logical properties of these. As each complex concept learned by the model corresponds to a specific logical relation, we tallied the total valid relations associated with a single image. These relations encompassed the AND, OR, and NOT relations of the three attributes constituting the image, amounting to a total of 66 relations per image. Subsequently, within the top 66 concepts retrieved with the same retrieval procedure described above, we count as hit only the ones that are true for the evaluated image. We called this task Logical Pattern Recognition (LPR). We conducted the LPR task across multiple iterations, systematically altering the top-k parameter for concept retrieval, ranging from 1 to 66—the maximum count of true relations for each image. This variation enables a comprehensive assessment of the model's capability to accurately retrieve representations. A qualitative example of LPR is shown in Figure 1.

4.2.1 Results

The results of the LPR task are visible in Figure 2. As illustrated in Figure (2a), the model achieves higher scores at lower k values compared to the complete set of valid representations. This implies that, even if the model doesn't precisely match all valid relations for each image, it still performs well on a subset of them.

Furthermore, our observations reveal that scores for AND patterns are consistently lower than those for other patterns. This discrepancy can be attributed to the fact that logical OR and NOT relations impose less stringent constraints on the images they must match to be deemed valid. The AND pattern is validated only when the target image contains both attributes referenced by the relation, while the OR pattern requires the image to contain at least one of the two related attributes. In contrast, the NOT pattern only demands the absence of the attribute it

Table 3: Top 3 Accuracy Scores for Logical Pattern Recognition by operator (higher is better).

Top-k Num	Split	Tot	NOT	AND	OR
10	D_{test_no}	0.8682	0.9305	0.5667	0.9201
	D_{test_v}	0.8827	0.9364	0.5878	0.9158
20	D_{test_no}	0.8411	0.9344	0.5326	0.8935
	D_{test_v}	0.8435	0.9261	0.5516	0.8809
30	D_{test_no}	0.8185	0.9201	0.4900	0.8738
	D_{test_v}	0.8179	0.9248	0.5025	0.8599
40	D_{test_no}	0.7957	0.9137	0.4444	0.8533
	D_{test_v}	0.7884	0.9150	0.4478	0.8348
50	D_{test_no}	0.7621	0.9057	0.3776	0.8259
	D_{test_v}	0.7543	0.9081	0.3965	0.8035
60	D_{test_no}	0.7157	0.8990	0.3136	0.7847
	D_{test_v}	0.7052	0.8960	0.3276	0.7605
66	D_{test_no}	0.6830	0.8893	0.2794	0.7538
	D_{test_v}	0.6712	0.8906	0.2919	0.7294

negates in the target image.

To provide a comprehensive overview of the model's performance, we present a breakdown of scores across the two evaluation settings and the three logical pattern categories—AND, OR, and NOT—at six evaluation points, along with a cumulative total. Refer to Table 3 for a detailed account of these scores.

We proceeded to delve deeper into the underlying factors contributing to the low scores observed in AND patterns. We analyzed the error rate contributions of the three distinct categories of AND patterns: Color AND Material, Color AND Shape, and Material AND Shape. As we can observe in Figure 2b, throughout the evaluation process, the error contribution from Color AND Shape remained negligible in the initial trials but exhibited an upward trend as the number of representations retrieved increased. The elevated error rate in the early trials can then be attributed to the other two categories, namely Color AND Material and Material AND Shape. Notably, considering the lower scores for materials in Multi-Attributes-Recognition, we can infer that the model's low performance in adequately representing this attribute category may be a key factor contributing to its diminished performance in the AND pattern task.

5 Conclusion

One major difficulty of this work consisted in training the multi-task network: hyper-networks grow exponentially in size (our network counted 29M parameters, compared to the baseline of 1.6M parameters), depending on the network they are abstracting from the task, which can dramatically slow down training. Moreover, due to their inductive bias, interpolation between task-specific weights can lead to concurrent objectives and thus complications such as catastrophic forgetting. We conducted early experiments on recent and promising approaches such as modular, composable skills [19], observing that comparable performance to hyper-networks is achieved; we consequently aim at expanding our experiments on complex concepts to this

new type of architecture.

We then demonstrate the capability of comparative learning in enabling models to represent more complex compound concepts compared to single attribute concepts. To further explore this, we propose investigating how these learned complex representations could enhance performance in tasks such as image retrieval. This improvement can be achieved by strengthening the effectiveness of logical expressions used as queries.

A subsequent challenge consists of unifying the learning of multiple complex compound concepts with a single multi-task model. We hypothesize that prior knowledge of primitive concepts could ease the learning of compositions of these, specifically logical expressions for which the semantics of the employed operators are additionally learned. In practice, we would compare the achieved performance against the sample efficiency of a pre-trained "primitive" model fine-tuned on logical expressions, with respect to a second model trained on such logical compositions from scratch.

Acknowledgments

This work is the result of the final project of the Grounded Language Processing course held by Professor Raffaella Bernardi[12], raffaella.bernardi@unitn.it and Professor Paolo Rota [12], paolo.rota@unitn.it.

References

[1] Josh Abramson, Arun Ahuja, Federico Carnevale, Petko Georgiev, Alex Goldin, Alden Hung, Jessica Landon, Jirka Lhotka, Timothy Lillicrap, Alistair Muldal, George Powell, Adam Santoro, Guy Scully, Sanjana Srivastava, Tamara von Glehn, Greg Wayne, Nathaniel Wong, Chen Yan, and Rui Zhu. Improving multimodal interactive agents with reinforcement learning from human feedback, 2022.

[2] Saeed Amizadeh, Hamid Palangi, Oleksandr Polozov, Yichen Huang, and Kazuhito Koishida. Neuro-symbolic visual reasoning: Disentangling "visual" from "reasoning", 2020.

[3] Yuwei Bao, Barrett Lattimer, and Joyce Chai. Human inspired progressive alignment and comparative learning for grounded word acquisition. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15475–15493, Toronto, Canada, July 2023. Association for Computational Linguistics.

[4] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline, 2020.

[5] Vinod Kumar Chauhan, Jiandong Zhou, Ping Lu, Soheila Molaei, and David A. Clifton. A brief review of hypernetworks in deep learning.

[6] Liangyu Chen, Bo Li, Sheng Shen, Jingkang Yang, Chunyuan Li, Kurt Keutzer, Trevor Darrell, and Ziwei Liu. Large language models are visual reasoning coordinators, 2023.

[7] Michael Crawshaw. Multi-task learning with deep neural networks: A survey, 2020.

[8] Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R. Arabnia. A brief review of domain adaptation, 2020.

[9] Dedre Gentner. Structure-mapping: A theoretical framework for analogy. Cognitive Science, 7(2):155–170, 1983.

[10] Dedre Gentner and Arthur B. Markman. Structural alignment in comparison: No difference without similarity. Psychological Science, 5(3):152–158, 1994.

[11] Susan J. Hespos, Erin Anderson, and Dedre Gentner. Structure-Mapping Processes Enable Infants' Learning Across Domains Including Language, pages 79–104. Springer International Publishing, Cham, 2020.

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[13] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. Foundations and Trends® in Machine Learning, 12(4):307–392, 2019.

[14] Satwik Kottur, José M. F. Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. Clevr-dialog: A diagnostic dataset for multi-round reasoning in visual dialog, 2019.

[15] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2023.

[16] Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks, 2021.

[17] Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. A mechanism for solving relational tasks in transformer language models, 2023.

[18] Jonas Pfeiffer, Sebastian Ruder, Ivan Vulić, and Edoardo Maria Ponti. Modular deep learning, 2024.

[19] Edoardo M. Ponti, Alessandro Sordoni, Yoshua Bengio, and Siva Reddy. Combining modular skills in multitask learning, 2022.

[20] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[21] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks, 2022.

[22] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application, 2024.

[23] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bag-of-words models, and what to do about it? arXiv preprint arXiv:2210.01936, 2022.

[24] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it?, 2023.

[25] Tian Yun, Usha Bhalla, Ellie Pavlick, and Chen Sun. Do vision-language pretrained models learn composable primitive concepts?, 2023.