Im-Promptu: In-Context Composition from Image Prompts

Bhishma Dedhia[1], Michael Chang[2], Jake C. Snell[3], Thomas L. Griffiths[3,4], Niraj K. Jha[1]

[1]Department of Electrical and Computer Engineering, Princeton University
[2]Department of Computer Science, University of California Berkeley
[3]Department of Computer Science, Princeton University
[4]Department of Psychology, Princeton University
{bdedhia,js2523,tomg,jha}@princeton.edu, mbchang@berkeley.edu

Abstract

Large language models are few-shot learners that can solve diverse tasks from a handful of demonstrations. This implicit understanding of tasks suggests that the attention mechanisms over word tokens may play a role in analogical reasoning. In this work, we investigate whether analogical reasoning can enable in-context composition over composable elements of visual stimuli. First, we introduce a suite of three benchmarks to test the generalization properties of a visual in-context learner. We formalize the notion of an analogy-based in-context learner and use it to design a meta-learning framework called *Im-Promptu*. Whereas the requisite token granularity for language is well established, the appropriate compositional granularity for enabling in-context generalization in visual stimuli is usually unspecified. To this end, we use *Im-Promptu* to train multiple agents with different levels of compositionality, including vector representations, patch representations, and object slots. Our experiments reveal tradeoffs between extrapolation abilities and the degree of compositionality, with non-compositional representations extending learned composition rules to unseen domains but performing poorly on combinatorial tasks. Patch-based representations require patches to contain entire objects for robust extrapolation. At the same time, object-centric tokenizers coupled with a cross-attention module generate consistent and high-fidelity solutions, with these inductive biases being particularly crucial for compositional generalization. Lastly, we demonstrate a use case of *Im-Promptu* as an intuitive programming interface for image generation.

1 Introduction

> No thought can be formed that isn't informed by the past; or, more precisely, we think only thanks to analogies that link our present to our past.
>
> D. Hofstadter and E. Sander, Surfaces and Essences [1]

Humans represent complex concepts by combining and organizing simpler concepts [2, 3]. This endows us with an uncanny skill of constructing an unlimited number of concepts by composing a relatively small set of previously learned basic building blocks, an ability more formally called *compositional generalization*. For example, we can generate sentences by combining a dictionary of words in different ways and solve a wide range of mathematical problems using a small set of basic arithmetic operations and variables. In the case of the visual world, objects form naturally

composable entities and object-centric primitives can be flexibly combined using abstract syntactic rules to yield novel and counterfactual scenes.

The growing body of work in object-centric learning methods [4, 5, 6] enable the extraction of latent object representations from perceptual scenes. However, slot composition methods have been limited to *ad-hoc* composition rules based on hard-coding slots into concept libraries [7, 8]. A central challenge in learning a generative model that explicitly models the compositional grammar of the visual world is that the seemingly unlimited number of composition rules that undergird real-life visual entities prevents the *explicit* specification of each rule. For example, a warm spring day consists of blooming flowers and freshly verdant expanses, but come autumn, the same scene stipulates the presence of golden meadows carpeted with fallen leaves. State-of-the-art text-to-image models [9, 10, 11, 12] circumvent this bottleneck by grounding images in natural language. This aligns the representations of entities in an image to the compositional structure of language. While the visual quality of these models is stunning, they often require engineering complex language prompts to elicit the desired output. They also suffer from limited understanding of object relations [13]. Moreover, studies [14] have shown that infants as young as six months old can extract compositional structure from sequences of visual events, even before they acquire language. Such compositional abstraction independent of language faculty exhibited by humans remains elusive for contemporary models.
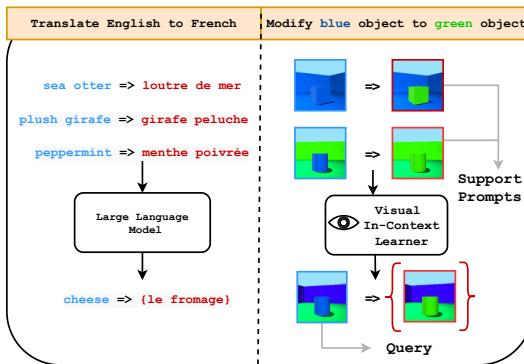

Figure 1: LLMs perform in-context understanding of a task from a few examples in an analogical manner (left). We use analogy-making to implicitly understand the composition rules over visual stimuli made of object-like entities (right).

Large language models (LLMs) [15, 16, 17, 18, 19, 20], on the other hand, demonstrate *implicit* composition skills. Pre-trained on large text corpora, these models have impressive emergent few-shot learning capabilities. They can solve novel tasks on the fly without gradient updates by following examples shown in the instruction prompts, a process known as *in-context learning*. Under the hood, in-context learning uses the inherently compositional nature of the input text to implicitly infer the task structure and how to apply the task structure to a query [21]. In the most simple form, the prompt-based few-shot learning paradigm can be formalized as *analogy solving* of the type $A : B :: C : D$, where A represents an example task, B represents the example solution, and C denotes the query. The generated solution is the analogy completion D. Analogies over compositional elements, therefore, provide a direct isomorphism between in-context learning for language and learning implicit composition rules for visual objects, as depicted in Fig. 1. While LLMs use pre-specified tokenizers [22, 23, 24], the compositional granularity for *implicit* visual understanding is unspecified. On the other hand, slot methods induce object-level representations but fail to learn composition rules. In this work, we then ask:

Can analogy-solving enable in-context generalization over object-centric compositional elements of visual entities?

While many linguistic corpora have been extracted from the Internet for text-based tasks, to the best of our knowledge, no datasets exist for measuring in-context composition capability from visual prompts. To this end, we introduce a suite of three diverse benchmarks in Section 3 built on top of rich combinatorial spaces to systematically test image-based in-context learning abilities: (a) 3D Shapes [25], (b) BitMoji Faces, and (c) CLEVr Objects [26]. If the attention over word tokens enables analogical understanding in LLMs, can we transfer this mechanism to visual analogies? We formulate a unifying framework in Section 4.1 to formalize this notion. We present a visual analogy-based meta-learning algorithm called *Im-Promptu* in Section 4.2 and use it to train agents with a cross-attention module. How does one tokenize visual inputs? To answer this, we model several generative agents in Section 6, encompassing a continuum ranging from strong compositional learners using object-centric abstractions to non-compositional monolithic representation learners. Extensive experiments presented in Section 7 show the dependence of robust in-context learning for visual stimuli on object-centric slots as visual tokenizers and cross-attention. We demonstrate the use

of our visual analogy framework in task extrapolation (Section 7.2), combinatorial generalization (Section 7.3), and generating counterfactual images (Section 7.5).

2 Related Work

In this section, we discuss prior works and draw connections to relevant research areas.

2.1 Object-Centric Learning: This growing body of work targets unsupervised decomposition of scenes into a set of object-centric vectors from raw perception data. A common approach involves modeling an autoencoder consisting of a latent factorization of object slots that are independently decoded to reconstruct the scene [4, 5, 6, 27, 28, 29, 30, 31, 32]. These methods effectively segment scenes into representations of underlying objects but do not learn the flexible combination of object representations. On the other hand, the authors in [7] replace the independent slot decoding bias with an Image-GPT [33] and demonstrate novel slot composition. While this work largely answers how to compose a *given* set of visual entities, their slot selection is performed manually through organization of slots into clustered libraries. In our work, we use a meta-learning approach to help the learner implicitly understand the composition task. Recently, the authors of [34] recast the iterative slot refinement problem as finding of fixed points through implicit differentiation to improve the stability and computational cost of the slot-attention (SA) technique. The authors of [35] use specific inductive biases to disentangle object contents from spatial information that enables relational reasoning. Our work produces a more generalized composition beyond spatial relations.

2.2 In-Context Learning in LLMs: LLMs [15, 16, 17, 18, 19, 20] can perform in-context learning at inference time from natural language prompts. In this work, we aim to extend this paradigm beyond language without language grounding. Empirical studies [36, 37] have found that the success of in-context learning critically hinges on idiosyncrasies in training distribution, prompt text structure, and label examples. In contrast, theoretical interpretations [38] have formulated it as an instance of implicit Bayesian inference. Prior works [39, 40] have also attempted to improve language models by meta-learning on an in-context learning objective.

2.3 Analogical Reasoning: The ability to make analogies has been shown to have important implications in various domains, such as problem-solving, creativity, learning, and counterfactual thinking [41, 42, 43, 44, 1]. Various models have been proposed to explain the underlying mechanisms of analogy-making. Popular symbolic approaches include ARGUS [45], Transfer Frames [46], and Structural Mapping Engine [47], connectionist models include LISA [48] and Star-1 [49], and models that occupy a middle ground like Copycat [50]. Contemporary deep-learning models [51, 52, 53] attempt to solve Raven's Progressive Matrices and Bongard Problems, types of visual analogies present in IQ tests. The authors of [54] propose a generative deep visual analogy-making model based on a parallelogram regularizer on the latent space to disentangle variation factors. Besides, several works [55, 56] show emergent analogical structure in language models.

2.4 Compositional Generative Models: Prior works [57, 58] use energy-based models to represent composable factors of variation of a scene. However, these methods use concept labels [59] or are grounded in text and are limited to a few concept composition operations. A follow-up work [60] describes unsupervised energy-based concepts; however, the re-composition is performed via manually-picked concepts. Recent works in text-to-image models use language grounding to induce compositionality. DALL-E [9] uses language modeling over the joint space of the text and image tokens to facilitate image generation. The latest text-to-image models [10, 11, 12] use diffusion modeling to guide the image generated from the text. A growing body of work [61, 62, 63] explores text-grounded controllable image editing via prompt engineering, where most parts of the image are preserved, but specific objects are re-composed.

3 Benchmarks

Humans can contextualize past visual stimuli in the face of everyday experiences and even use them for imagining counterfactuals. Having seen an oak table, we can, with little difficulty, look at a metal armchair and understand the concept of an oak armchair. How do we then systematically test such abilities of a visual in-context learner? In this work, we posit that an answer to this question lies in combinatorial visual spaces with controllable factors of variation, which provide a *semantically rich, simple, scalable, and composable* avenue for setting up such tasks. We create a suite of three benchmarks (Fig. 2) from compositional image creators that include (a) 3D Shapes [25], (b) BitMoji
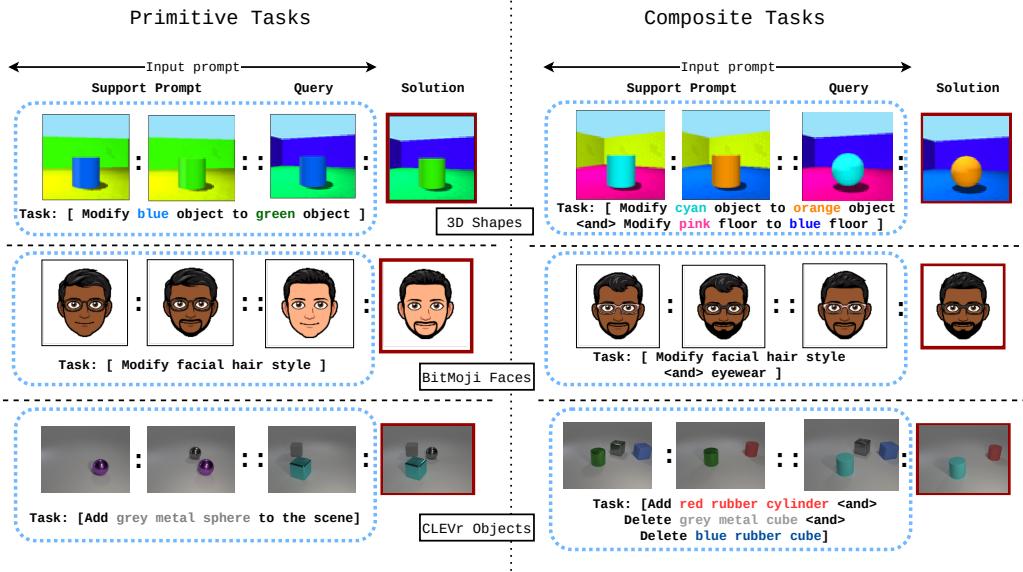
Figure 2: We use combinatorial image spaces to set up visual in-context learning tasks. Each task requires the generation of a solution from a query and supporting examples. The latent description of the task is described within the square parentheses. Left: Primitive tasks modify a composable element in isolation. Right: Composite tasks combine primitives to test the combinatorial generalization skills of the learner.

Faces, and (c) CLEVr Objects [26]. Each benchmark has a split of primitive training tasks and out-of-distribution test tasks. The *primitive* tasks (Fig. 2, left) constitute an analogy where a *single* composable element of the visual stimulus is modified in isolation. The description of each benchmark with its primitive set of tasks is given next:

3.1 3D Shapes Image Prompts: This dataset consists of static scenes of various objects lying on a colored floor in front of a colored wall viewed from different orientations. To set up primitive tasks, we consider the four properties $P = \{$object color, wall color, floor color, scene orientation$\}$, where each property can take multiple values within a domain. For each such task, the latent relation $r(.)$ for the analogy prompt takes a property $p \in P$ and modifies it from a *source* domain value to a *target* value. This benchmark comprises 80000 such tasks with a maximum of four supporting image pairs for each instance and with a roughly equal split across properties.

3.2 BitMoji Faces Image Prompts: BitMoji is an avatar creator service for social media users that allows them to create intricate cartoon faces. We queried the BitMoji API [64] to collect faces with four underlying dynamic elements $P = \{$skin tone, facial hair style, hair style, eyewear$\}$. Much like the previous case, primitive tasks are generated by intervening on a *source* value of a property and modifying it to a *target* value. We populate the training set with 80000 tasks, with four demonstrations available per task.

3.3 CLEVr Objects Image Prompts: CLEVr is a popular visual question-answering dataset with the visual component consisting of multiple objects lying in a scene. We use the CLEVr rendering engine to set up primitive tasks that include adding and deleting the same object across various scenes. In this fashion, we generate 60000 tasks with three examples per task.

A composable image space allows the concurrent modification of individual visual elements. One can construct *composite* tasks (Fig. 2, right) out of the combination of primitives shown in the training set. A k-composite task is denoted as $R^k(.) = r_1 \circ r_2 \circ \cdots \circ r_k$ where $r_1, \cdots, r_k \in \mathcal{R}$. We provide details of the k-compositeness test of each of our agents in Section 7.3. Finer details about each benchmark, along with visual examples from every task, are detailed in Appendix A. We will release the complete dataset of primitive and composite tasks upon the publication of this article.

4

4 Methods

To learn compositional structure over visual stimuli, we begin by formulating a unified mechanistic interpretation of an in-context compositional learner using insights from design elements of LLMs.

4.1 In-Context Learning as Analogy Completion: A general *compositional* in-context learner $\mathcal{M}_{\phi,\alpha,\theta}(.)$ can be formalized via the following three key components:

- An encoder $\mathcal{E}_\phi(.)$ that maps the input space $\mathcal{X}$ to the compositional space $\mathcal{C} \triangleq \mathcal{E}_\phi(.) : \mathcal{X} \longmapsto \mathcal{C}$
- An executor $\mathcal{T}_\alpha(.,.,.)$ that maps compositional entities from example tasks, example solutions, and a query task to the query solution $\triangleq \mathcal{T}_\alpha(.,.,.) : \mathcal{C} \times \mathcal{C} \times \mathcal{C} \longmapsto \mathcal{C}$
- A decoder $\mathcal{D}_\theta(.)$ that maps the compositional space back to the input space $\triangleq \mathcal{D}_\theta(.) : \mathcal{C} \longmapsto \mathcal{X}$

Coupling the above functions together yields the learner:

$$\mathcal{M} \triangleq \mathcal{D}_\theta \left(\mathcal{T}_\alpha \left(\mathcal{E}_\phi(.), \mathcal{E}_\phi(.), \mathcal{E}_\phi(.) \right) \right) : \mathcal{X} \times \mathcal{X} \times \mathcal{X} \longmapsto \mathcal{X} \tag{1}$$

The inclusion of $\mathcal{E}_\phi(.)$ and $\mathcal{D}_\theta(.)$ is crucial for compositional abstraction since the input space may not be *a priori* composable as in the case of images. Beyond this, we do not place any strong parametric constraints on the model. Let $\mathcal{R}$ denote a task set. For any task $r \in \mathcal{R}$, define the task analogy ζ over input pairs x_1, x_2 as $x_1 : r(x_1) :: x_2 : r(x_2) \triangleq A : B :: C : D$. An effective in-context learner over $\mathcal{R}$ then satisfies the following property [65]:

$$\forall r \in \mathcal{R}, \ \mathbb{E}_{\zeta \sim \mathcal{P}(r,\mathcal{X})} \left[\mathcal{L} \left(D; \mathcal{M}_{\phi,\alpha,\theta}(A,B,C) \right) \right] \leq \epsilon \tag{2}$$

Here, $\mathcal{P}$ denotes the distribution of the analogies resulting from the input space $\mathcal{X}$ and task $r(.)$, $\mathcal{L}(.)$ denotes a loss metric, and ϵ is an error bound. The above property simply states that a good in-context learner solves analogies defined over a task set within some finite error bound. In Appendix B, we describe LLMs as an instantiation of this framework.

4.2 Im-Promptu Learning: Pre-training LLMs on large-scale text data leads to the emergence of highly transferable internal representations that allow them to adjust to novel tasks by simple priming on a few demonstrations. However, it is still unclear whether the underlying structure of the training data, architecture, or inductive biases instilled by choice of pre-training learning algorithms cause these intriguing properties to emerge. How can we encourage the emergence of such a general-purpose representation for visual data? In this work, we take an explicit approach based on the formalism presented in the previous section. Having established the desideratum (Eq. 2), we set up a learning procedure called *Im-Promptu* that trains the model to generate analogy completions from image prompts. To this end, given a task set $\mathcal{R}$, in each minibatch, we sample an analogy $A : B :: C : D$ that follows a latent primitive $r(.) \in \mathcal{R}$. Given a loss criterion, $\mathcal{L}$, we make the following stochastic gradient updates of the model parameters:

$$\{\phi^{t+1}, \alpha^{t+1}, \theta^{t+1}\} = \{\phi^t, \alpha^t, \theta^t\} - \frac{\alpha}{N} \sum_{i=1}^{N} \nabla_{\theta^t, \alpha^t, \phi^t} \mathcal{L} \left(D, \mathcal{M}_{\phi^t, \alpha^t, \theta^t}(A,B,C) \right) \tag{3}$$

where the step size α is a hyperparameter. The full *Im-Promptu* algorithm is laid out in Algorithm 1.

Algorithm 1 Im-Promptu Learning Algorithm

Require: Task set $\mathcal{R}$, Step Size α
Initialize parameters θ, α, ϕ of model $\mathcal{M}$
while not done **do**
    Sample primitive task $r(.) \sim \mathcal{R}$
    Sample N input image pairs $X_1^{1:N}, X_2^{1:N} \sim \mathcal{P}_{train}$
    for i in $1 \cdots N$ **do**
        Sample analogy $\zeta^i \triangleq x_1^i : r(x_1^i) :: x_2^i : r(x_2^i) \triangleq A : B :: C : D$
    end for
    Update $\{\theta, \alpha, \phi\} = \{\theta, \alpha, \phi\} - \frac{\alpha}{N} \nabla_{\theta,\alpha,\phi} \sum_i \mathcal{L}_{\mathcal{M}}(\zeta^i)$
end while

---

5 Background

We give brief details of the Slot Attention Mechanism and Slot Attention Transformer (SLATE) that form essential components of our best-performing agents.

5.1 Slot Attention: Object-Centric Learning frameworks decompose scenes into compositional object files called slots. SA [6] is a powerful autoencoder-based inductive bias for learning such slots. The encoder $f_\phi(.)$ initializes a set of N symmetric and independent slots $S_{1:N}^0 \triangleq \{s_1^0, s_2^0, \cdots, s_N^0\}$ that iteratively compete over T timesteps to attend to the input scene X and break symmetry to yield the final slots $S_{1:N}^T \triangleq \{s_1^T, s_2^T, \cdots, s_n^T\}$. A decoder $g_\theta(.)$ then individually decodes each slot in the final set into the pixel space to yield N images $I_{1:N}$ and their corresponding masks $M_{1:N}$. A convex combination of the images, using masks as weights, reconstructs the scene $\hat{X}$.

$$S_{1:N}^T = f_\phi(S_{1:N}^0, X) \tag{4}$$

$$I_1, M_1 = g_\theta(s_1^T), \cdots, I_N, M_N = g_\theta(s_N^T) \tag{5}$$

$$\hat{X} = M_1 \times I_1 + \cdots + M_N \times I_N \tag{6}$$

5.2 Slot Attention Transformer: Recent work [7] solves the pixel independence problem in SA [6] by learning a latent discrete library from pixels using a discrete variational autoencoder (dVAE) [66, 67] and then running SA over the vocabulary space of the dictionary. An imageGPT decoder [33] learns to predict the sequence of image latent from the slot prompts and, as a result, enables novel composition of slots. We lay out the forward pass of the autoencoder architecture below but defer the finer details to the original article.

$$Z_{1:L} = \{z^1, \cdots, z^L\} = f_\phi^{dVAE}(X) \rightarrow \text{ dVAE encoder} \tag{7}$$

$$S_{1:N}^T = SA_\phi(Z_{1:L}, S_{1:N}^0) \rightarrow \text{Slot Attention} \tag{8}$$

$$\hat{Z}_{1:L} = g_\theta^{GPT}(S_{1:N}^T) \rightarrow \text{ImageGPT} \tag{9}$$

$$\hat{X} = g_\theta^{dVAE}(\hat{Z}_{1:L}) \rightarrow \text{dVAE decoder} \tag{10}$$

Here, X is the input, $Z_{1:L}$ is the latent discrete sequence of length L, and $S_{1:N}$ denotes the N slots.

6 Learning Agents

In this section, we explore various modeling choices for a visual in-context learner ranging from simple pixel-space rules to object-centric learners (OCLs).

6.1 Pixel Baseline: This non-parametric baseline manipulates the pixel space using a simple addition rule. We denote the support prompt as $A : B$ and the query as C, the solution $\hat{D}$. Then the Pixel baseline predicts $\hat{D}$ as the outcome of a linear transformation, $\hat{D} = C + (B - A)$.

6.2 Monolithic Learner: Our first learning agent, inspired from [54], is entirely non-compositional and uses latent monolithic vectors to execute the image prompts as follows:

$$\mathcal{V} = f_\alpha\left([e_\phi(A), e_\phi(B)]\right), \ \hat{D} = d_\theta\left(h_\beta\left([\mathcal{V}, e_\phi(C)]\right)\right) \tag{11}$$

The encoder $e_\phi(.)$ and decoder $d_\theta(.)$ are convolutional and deconvolutional networks, respectively. The inference network $f_\alpha(.)$ and executor network $h_\beta(.)$ are modeled as multi-layered perceptrons. Here, $[\ ]$ denotes the concatenation operator. This architecture is detailed in Appendix C.1.

6.3 Object-Centric Learner (OCL): This agent, as depicted in Fig. 3, is a strongly compositional learner that uses object-centric inductive biases. The encoder and decoder of the framework in Section 4.1 are parametrized using SLATE [7] (see Section 5.2) as follows:

$$\mathcal{E}_\phi(.) \triangleq SA_\phi(f_\phi^{\text{dVAE}}(.)), \ \mathcal{D}_\theta(.) \triangleq g_\theta^{dVAE}(g_\theta^{GPT}(.)) \tag{12}$$

A Context Encoder Transformer (CET) $\triangleq \mathcal{T}_\alpha(., ,. ,.)(.)$ with interleaved layers of: (a) cross-attention on the context slots and (b) self-attention, induces sparse modifications on the query slots to complete the analogy.

$$S_{1:N}^A = \mathcal{E}_\phi(A), \ S_{1:N}^B = \mathcal{E}_\phi(B), \ S_{1:N}^C = \mathcal{E}_\phi(C) \tag{13}$$

$$S_{1:N}^D = CET\left(\text{query} = S_{1:N}^C, \text{keys, values} = S_{1:N}^A, S_{1:N}^B\right) \tag{14}$$
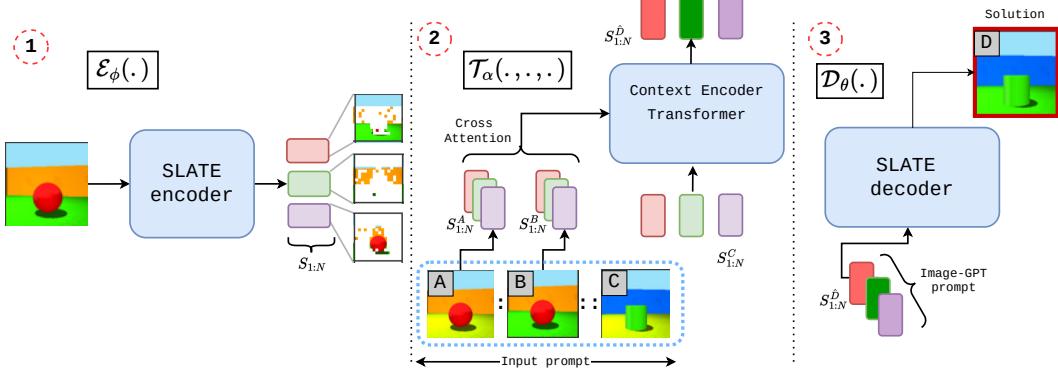
Figure 3: The OCL uses compositional object vectors to answer image prompts. (**1**) The encoder $\mathcal{E}_\phi(.)$ is a pre-trained SLATE [7] encoder that uses a dVAE to obtain a latent discrete sequence for the input image and runs SA [6] over the sequence to extract object-centric slots. (**2**) The CET modifies query slots $S_{1:N}^C$ via cross-attention over the support prompt slots $S_{1:N}^A, S_{1:N}^B$. (**3**) The modified slots are used to prompt an Image-GPT [33] that outputs the latent discrete sequence of the task solution D that is decoded by the dVAE decoder.

The latent sequence $\hat{Z}_D$ predicted by Image-GPT is mapped by the dVAE back to the pixel space $\hat{D}$.

$$\hat{Z}_D = g_\theta^{GPT}(S_{1:N}^D), \; \hat{D} = g_\theta^{dVAE}(\hat{Z}_D) \tag{15}$$

The CET forward pass is shown in Appendix C.2.

6.4 Sequential Prompter: This (Seq., visualized in Appendix C.3) is an ablation over the OCL that replaces CET with an LLM-like sequence prompt obtained from a simple concatenation of slots of images A, B, and C, injected with position embeddings p_i.

$$\mathcal{T}_\alpha(.,.,.) \triangleq [S_{1:N}^A, S_{1:N}^B, S_{1:N}^C] + p_i \tag{16}$$

While the OCL explicitly encodes the entire context into a fixed-length slot sequence via the CET, the ablated agent models a longer concatenated sequence.

6.5 Patch Learner: This agent straddles the compositionality extremes of Monolithic Learners and OCLs, and uses image patch abstractions. The SA module from OCL is ablated to get a discrete sequence of 4×4 patch latents as the degree of compositional granularity $\mathcal{E}_\phi(.) \triangleq f_\phi^{dVAE}(.)$. A larger image implies longer sequences for this modeling choice.

7 Experiments

In this section, we set up experiments to answer (1) whether analogical reasoning enables in-context generalization, (2) does such a generalization require key mechanisms from LLMs, namely compositionality and attention, (3) if compositionality is a key ingredient, what is the correct granularity for visual tokens, and (4) what are the limits to the generalization.

7.1 Training Setup: We trained each agent on the primitive set of tasks across the three benchmarks using Im-Promptu (Section 4.2). The SA-based object-centric encoders used in the OCL and Sequential Prompter were pre-trained as a SLATE autoencoder, as described by the authors in [7]. On the other hand, the Patch Learner was trained completely from scratch. The loss metric $\mathcal{L}$ used to train the agents was cross-entropy (CE) loss between the true latent sequence Z_D and the predicted solution $\hat{Z}_D$ obtained from the Image-GPT, i.e., $\mathcal{L}_{impromptu} = CE(Z_D, \hat{Z}_D)$.

In addition to the above loss, the dVAE was trained using the mean-squared error (MSE) loss over the raw pixel space to yield the full loss function $\mathcal{L} = \mathcal{L}_{impromptu} + MSE(D, \hat{D})$. For inference, answers of the transformers-based agents were sampled from the Image-GPT decoder using top-k nucleus sampling [68]. Hyperparameters for training and inference have been laid out in Appendix D.

7.2 Primitive Task Extrapolation: In the first out-of-distribution paradigm, we tested the ability of agents to apply learned rules to different domain pairs. In order to do this, we held out 20% of source-target pairs (see Section 3 for the definition of source and target) from the training primitives and tested the ability of agents to generalize from learned rules to unseen source-target pairs. For
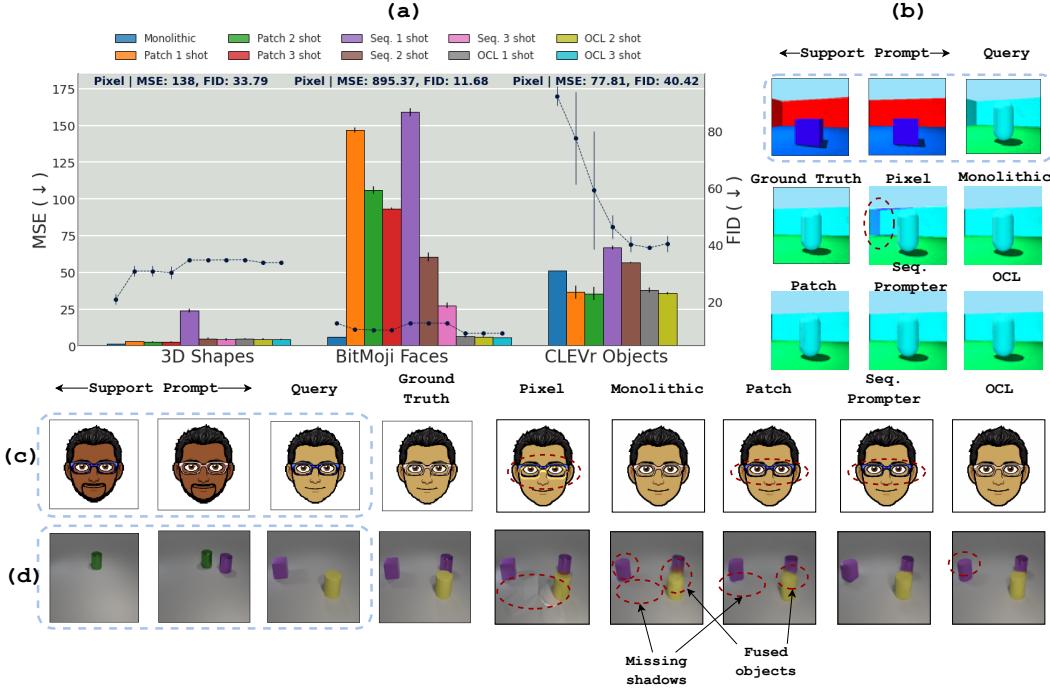
Figure 4: Primitive task extrapolation: (a) Plot showing scores where the left y-axis and the bars represent the MSE scores, while the right y-axis and the dotted line denote the FID scores. (b)-(d) Comparison of agent solutions with the dotted red circle representing anomalies with respect to the ground truth.

example, the `object color` property in the 3D Shapes benchmark can take 10 unique values and $10 \times 9 = 90$ source-target combinations. Thus, an agent was trained on only $90 \times 0.8 = 72$ pairs for object-hue primitives. We made sure that each value in the domain set was shown at least once as either the target or the source.

Fig. 4(a) plots scores of different agents across benchmarks against two key metrics: (1) MSE (lower is better) that quantitatively compares the construction against the ground truth and (2) Fréchet inception distance (FID, lower is better) score to measure the perceptual quality of the composition. We make several interesting observations:

(R1.1) Simple pixel manipulation produces implausible outputs. The baseline has poor MSE scores (~ 138 for 3D Shapes, ~ 895 for BitMoji) with clearly apparent artifacts in the output. While linear pixel transformations are sufficient when an object is independent of the remaining entities, it is unable to model complex dependencies (see Fig. 4(b),(d) under 'Pixel').

(R1.2) Monolithic Learners are reasonably effective at generalizing from learned rules on structured stimuli. These agents have the lowest MSE scores on 3D Shapes and BitMoji Faces, both spatially consistent datasets. However, the output deteriorated over CLEVr Objects where the latent scene configuration is diverse. The edges of the generated shapes and lighting details were significantly blurred, and occluded objects were often fused together (FID ~ 92, worse than Pixel baseline, Fig. 4(d) under 'Monolithic').

(R1.3) Patch Learner and Sequential Prompter benefit from additional examples. For these agents, the addition of context examples was markedly apparent from the significant MSE drops for > 1 shot (see Fig. 4(a)). We posit that since these models learn from longer contexts, additional examples are particularly effective in regularizing the output sequence manifold of the Image-GPT.

7.3 Composite Task Extrapolation: In this generalization paradigm, we probed the compositional generalization capabilities of the agents. A k-composite task ($k \geq 2$, see Section 3) is an unordered composite function of k primitives. The value of k is varied across each benchmark, where higher

8
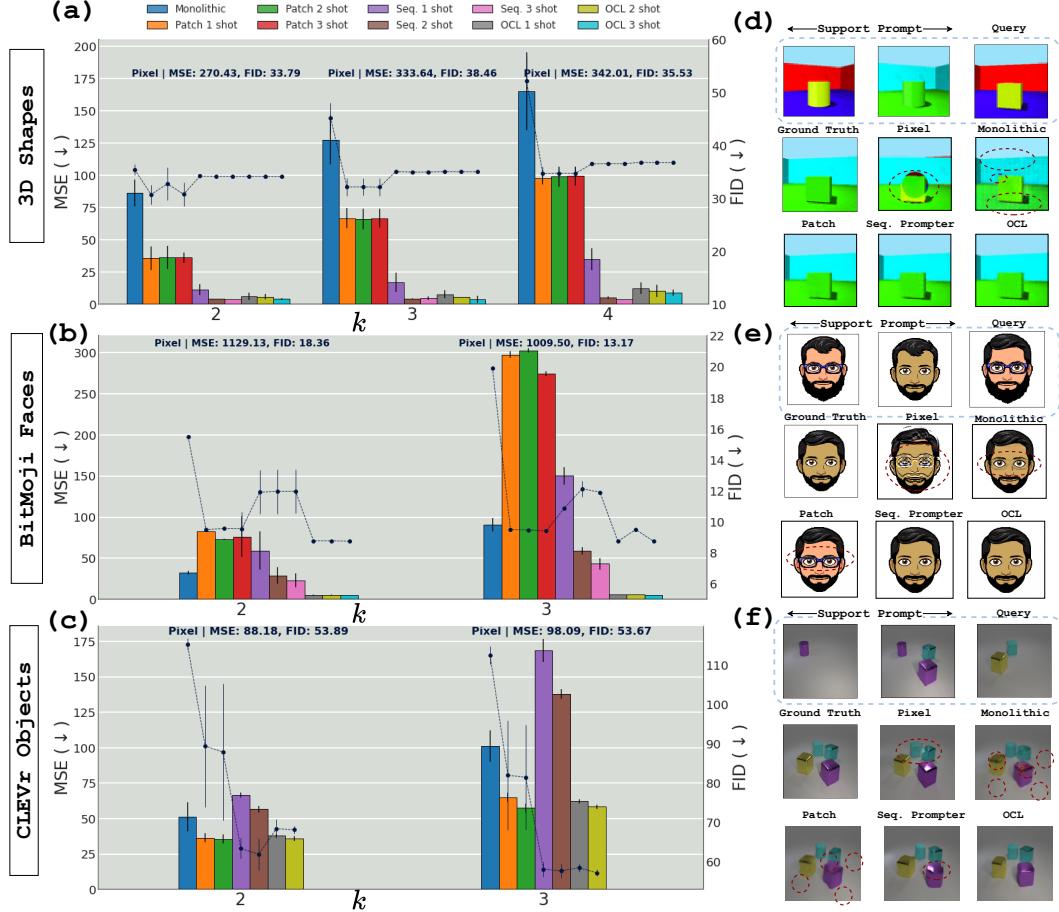
Figure 5: Composite task extrapolation: (a)-(c) Plots showing scores across benchmarks where the left y-axis and the bars represent the MSE scores, while the right y-axis and the dotted line denote the FID scores. k denotes the level of compositeness. (d)-(f) Comparison of agent solutions with the dotted red circle representing anomalies with respect to the ground truth.

values of k yield visually complex contexts. To solve such a task, the agent must identify the underlying primitive relations and combinatorially modify the query.

In Fig. 5(a), we note (R1.1) and (R1.3) from the previous setup in this extrapolation paradigm as well, but we make additional key observations next:

(R2.1) The effect of object-centric biases is strong. This is indicated by the widening MSE gap between the OCL and monolithic agent for increasing values of k across all three benchmarks (see Fig. 5(a)-(c)). Moreover, the OCL and the Sequential Prompter generate outputs with consistent perceptual quality (lower FID scores).

(R2.2) Monolithic and Patch Learners apply shortcuts. The monolithic agent tends to produce a Frankenstein-like image, as if a superposition of context images (see Fig. 5(d)-(f) under 'Monolithic'). In the case of the CLEVr Objects dataset, both the Monolithic and Patch Learners often simply filled in 'blobs' agnostic to the color and shape of objects (Fig. 5(f), Appendix E.3), with missing shadows leading to lower MSE scores but poor visual quality.

7.4 Analysis: Next, we discuss key elements required for in-context composition.

(A1) In-context generalization critically hinges on compositional granularity. As evident from (R1.2), monolithic representations suffice for reliably learning primitive composition rules over simple contexts. However, they suffer from fidelity issues when the visual space becomes more complex. Beyond primitive generalization, patch abstractions are effective when the underlying space has object-patch equivalence, as evident from the CLEVr Objects dataset (Fig. 5(c)). However, even

9
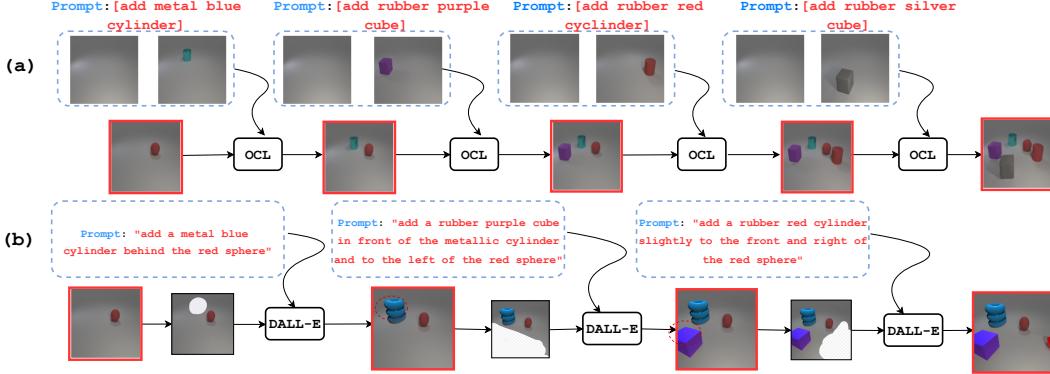
Figure 6: Scene creation from object components: (a) OCL trained via Im-Promptu is used to generate a scene of objects. Image-Prompts engineered using existing assets reliably generate the scene object by object. (b) The same scene is generated via language grounding and in-painting features of DALL-E [9]. Object properties are distorted and specifying the location of objects is tedious via language.

with object-patch equivalence, the Patch Learner omits the inter-object dependencies (e.g., object occlusions and shadows). It further fails to learn complex and diverse objects spanning multiple patches reliably (does poorly on the BitMoji Faces dataset with diverse facial components). Slot-based compositionality with the CET not only enables the generalization of the learned primitives but also enables combinatorial generalization to composite tasks that require abstracting out the composition rule over each entity in isolation as well as composing the resultant interplay between them to generate a globally consistent output.

(A2) Encoding the context via cross-attention leads to sample efficiency. Longer slot sequences in the Sequential Prompter require multi-shot contexts across all benchmarks. The sample inefficiency was particularly observed on the CLEVr Objects dataset that uses six slots and, hence, much longer sequences. Cross-attention via CET is essential for implicit inference, enabling the learner to encode specific objects from the context.

7.5 'Counterfactual Prompt Engineering' with Im-Promptu: LLMs have spurred interest in several priming techniques [69, 70] that can 'engineer' the model into following user instructions. While this natural-language-based programming language provides an intuitive interface for tasks that have a fundamental degree of language abstraction (writing code, summarization, symbolic manipulation, etc.), it is not *a priori* obvious that textual descriptions should also scaffold image generators. Coming up with natural-language prompts for complex visual entities is often tedious and unintuitive. Instead, explaining an image is naturally easier from its entities and composition over them[1]. To this end, Im-Promptu provides an avenue for generating counterfactuals via image prompts. In Fig. 6(a) we demonstrate the creation of a scene from its object components via the OCL. As a point of comparison, in Fig. 6(b), using image-inpainting and natural language prompts with DALL-E [9] yields an unreliable output with distorted objects. More practically, the OCL reduces human effort significantly in rendering images with desired properties by using existing visual assets.

8 Conclusion

This work investigated whether analogy-solving can enable in-context compositional generalization over visual entities. We first designed a new test suite. We probed if attention and compositionality, key ingredients in LLMs, also play a crucial role in the visual domain. We transported these ingredients from language to model several agents. Our experiments showed that non-compositional agents do not generalize well beyond primitive task extrapolation. Even with compositionality baked in, the performance of patch-based learners depends on idiosyncrasies in the visual structure. However, we found that object-centric biases consistently facilitate an implicit understanding of composition rules and generate outputs with global semantic consistency. Our ablation of the CET suggested that cross-attention plays a crucial role, just like language models, which posits the importance of analogy and attention in understanding the underpinnings of in-context learning. Future research

---

[1]"No one is an artist unless he carries his picture in his head before painting it, and is sure of his method and composition" – Claude Monet

challenges include collecting real-world primitives from photo-realistic graphic engines or human labels, improving object-centric inductive biases for real-life entities, and designing better prompting techniques. We hope that our work will spur further research on visual in-context learning.

Acknowledgments and Disclosure of Funding

References

[1] Douglas R. Hofstadter and Emmanuel Sander. *Surfaces and Essences: Analogy as the Fuel and Fire of Thinking*. Basic Books, 2013.

[2] Brenden M. Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua B. Tenenbaum. One Shot Learning of Simple Visual Concepts. *Cognitive Science*, 33, 2011.

[3] Brenden M. Lake, Tal Linzen, and Marco Baroni. Human Few-Shot Learning of Compositional Instructions. *CoRR*, abs/1901.04587, 2019.

[4] Christopher P. Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. MONet: Unsupervised Scene Decomposition and Representation. *CoRR*, abs/2303.08774, 2019.

[5] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Chris Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-Object Representation Learning with Iterative Variational Inference. *CoRR*, abs/1903.00450, 2019.

[6] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-Centric Learning with Slot Attention. *CoRR*, abs/2006.15055, 2020.

[7] Gautam Singh, Fei Deng, and Sungjin Ahn. Illiterate DALL-E Learns to Compose. *CoRR*, abs/2110.11405, 2021.

[8] Jindong Jiang, Fei Deng, Gautam Singh, and Sungjin Ahn. Object-Centric Slot Diffusion. *CoRR*, abs/2303.10834, 2023.

[9] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-Shot Text-to-Image Generation. *CoRR*, abs/2102.12092, 2021.

[10] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents. *CoRR*, abs/2204.06125, 2022.

[11] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *CoRR*, abs/2205/11487, 2022.

[12] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. *CoRR*, abs/2112.10752, 2021.

[13] Gary Marcus, Ernest Davis, and Scott Aaronson. A Very Preliminary Analysis of DALL-E 2. *CoRR*, abs/2202.13807, 2022.

[14] Fei Xu, Elizabeth S. Spelke, and Sydney Goddard. Number Sense in Human Infants. *Developmental Science*, 8(1):88–101, 2005.

[15] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. *CoRR*, abs/2005.14165, 2020.

[16] Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`, May 2021.

[17] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. PaLM: Scaling Language Modeling with Pathways. *CoRR*, abs/2204.02311, 2022.

[18] Opher Lieber, Or Sharir, Barak Lenz, and Yoav Shoham. Jurassic-1: Technical Details and Evaluation. https://uploads-ssl.webflow.com/60fd4503684b466578c0d307/61138924626a6981ee09caf6_jurassic_tech_paper.pdf, 2022.

[19] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann,

Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sänger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. *CoRR*, abs/2211.05100, 2022.

[20] OpenAI. GPT-4 Technical Report. *CoRR*, abs/2303.08774, 2023.

[21] Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. Larger Language Models do In-context Learning Differently. *CoRR*, abs/2303.03846, 2023.

[22] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. *CoRR*, abs/1508.07909, 2016.

[23] Mike Schuster and Kaisuke Nakajima. Japanese and Korean Voice Search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152, 2012.

[24] Taku Kudo and John Richardson. SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing. *CoRR*, abs/1808.06226, 2018.

[25] Chris Burgess and Hyunjik Kim. 3D Shapes Dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

[26] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. *CoRR*, abs/1612.06890, 2016.

[27] Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. On the Binding Problem in Artificial Neural Networks. *CoRR*, abs/2012.05208, 2020.

[28] Martin Engelcke, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner. GENESIS: Generative Scene Inference and Sampling with Object-Centric Latent Representations. *CoRR*, abs/1907.13052, 2019.

[29] Martin Engelcke, Oiwi Parker Jones, and Ingmar Posner. GENESIS-V2: Inferring Unordered Object Representations without Iterative Refinement. *CoRR*, abs/2104/09958, 2021.

[30] Thomas Kipf, Gamaleldin F. Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional Object-Centric Learning from Video. *CoRR*, abs/2111.12594, 2021.

[31] Gamaleldin F. Elsayed, Aravindh Mahendran, Sjoerd van Steenkiste, Klaus Greff, Michael C. Mozer, and Thomas Kipf. SAVi++: Towards End-to-End Object-Centric Learning from Real-World Videos. *CoRR*, abs/2206.07764, 2022.

[32] Gautam Singh, Yi-Fu Wu, and Sungjin Ahn. Simple Unsupervised Object-Centric Learning for Complex and Naturalistic Videos. *CoRR*, abs/2205.14065, 2022.

[33] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative Pretraining From Pixels. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR, 13–18 Jul 2020.

[34] Michael Chang, Thomas L. Griffiths, and Sergey Levine. Object Representations as Fixed Points: Training Iterative Refinement Algorithms with Implicit Differentiation. *CoRR*, abs/2207.00787, 2022.

[35] James C. R. Whittington, Rishabh Kabra, Loic Matthey, Christopher P. Burgess, and Alexander Lerchner. Constellation: Learning Relational Abstractions over Objects for Compositional Imagination. *CoRR*, abs/2107.11153, 2021.

[36] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? *CoRR*, abs/2202.12837, 2022.

[37] Stephanie C. Y. Chan, Adam Santoro, Andrew K. Lampinen, Jane X. Wang, Aaditya Singh, Pierre H. Richemond, Jay McClelland, and Felix Hill. Data Distributional Properties Drive Emergent In-Context Learning in Transformers. *CoRR*, abs/2205.05055, 2022.

[38] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An Explanation of In-context Learning as Implicit Bayesian Inference. *CoRR*, abs/2111.02080, 2021.

[39] Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. Meta-learning via Language Model In-context Tuning. *CoRR*, abs/2110.07814, 2021.

[40] Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. MetaICL: Learning to Learn in Context. *CoRR*, abs/2110.15943, 2021.

[41] J. P. Guilford. Creativity. *American Psychologist*, 5(9):444, 1950.

[42] Dedre Gentner. Structure-Mapping: A Theoretical Framework for Analogy. *Cognitive Science*, 7(2):155–170, 1983.

[43] Arthur B. Markman and Dedre Gentner. Structural Alignment in Analogy and Similarity. *American Psychologist*, 48(1):40, 1993.

[44] Keith J. Holyoak and Paul Thagard. *Mental Leaps: Analogy in Creative Thought*. MIT Press, 1995.

[45] W. R. Reitman. *Cognition and Thought: An Information Processing Approach*. John Wiley and Sons, 1965.

[46] Patrick H. Winston. Learning by Creatifying Transfer Frames. *Artificial Intelligence*, 10(2):147–172, 1978.

[47] Brian Falkenhainer, Kenneth D. Forbus, and Dedre Gentner. The Structure-Mapping Engine: Algorithm and Examples. *Artificial Intelligence*, 41(1):1–63, 1989.

[48] Jeffrey E. Hummel and Keith J. Holyoak. Distributed Representations of Structure: A Theory of Analogical Access and Mapping. *Psychological Review*, 104(3):427–466, 1997.

[49] Graeme S. Halford, William H. Wilson, Jian Guo, Ross W. Gayler, Janet Wiles, and J. E. M. Stewart. *Connectionist Implications for Processing Capacity Limitations in Analogies*, pages 363–415. Ablex Publishing, 1994.

[50] Douglas R. Hofstadter. *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. Basic Books, 1995.

[51] Stefan Depeweg, Constantin A. Rothkopf, and Frank Jäkel. Solving Bongard Problems with a Visual Language and Pragmatic Reasoning. *CoRR*, abs/1804.04452, 2018.

[52] Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. RAVEN: A Dataset for Relational and Analogical Visual rEasoNing. *CoRR*, abs/1903.02741, 2019.

[53] Felix Hill, Adam Santoro, David G. T. Barrett, Ari S. Morcos, and Timothy Lillicrap. Learning to Make Analogies by Contrasting Abstract Relational Structure. *CoRR*, abs/1902.00120, 2019.

[54] Scott E. Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. Deep Visual Analogy-Making. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[55] Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[56] Taylor Webb, Keith J. Holyoak, and Hongjing Lu. Emergent Analogical Reasoning in Large Language Models. *CoRR*, abs/2212.09196, 2022.

[57] Yilun Du, Shuang Li, and Igor Mordatch. Compositional Visual Generation with Energy Based Models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6637–6647. Curran Associates, Inc., 2020.

[58] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. Compositional Visual Generation with Composable Diffusion Models. *CoRR*, abs/2206.01714, 2022.

[59] Drew A. Hudson and C. Lawrence Zitnick. Compositional Transformers for Scene Generation. *CoRR*, abs/2111.08960, 2021.

[60] Yilun Du, Shuang Li, Yash Sharma, Joshua B. Tenenbaum, and Igor Mordatch. Unsupervised Learning of Compositional Energy Concepts. *CoRR*, abs/2111.03042, 2021.

[61] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-Prompt Image Editing with Cross Attention Control. *CoRR*, abs/2208.01626, 2022.

[62] Dani Valevski, Matan Kalman, Yossi Matias, and Yaniv Leviathan. UniTune: Text-Driven Image Editing by Fine Tuning an Image Generation Model on a Single Image. *CoRR*, abs/2210.09477, 2022.

[63] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-Based Real Image Editing with Diffusion Models. *CoRR*, abs/2210.09276, 2022.

[64] Snap Inc. Bitmoji. `https://www.bitmoji.com/`, accessed 2023.

[65] Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What Can Transformers Learn In-Context? A Case Study of Simple Function Classes. *CoRR*, abs/2208.01066, 2022.

[66] Jason Tyler Rolfe. Discrete Variational Autoencoders. *CoRR*, abs/1609.02200, 2016.

[67] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural Discrete Representation Learning. *CoRR*, abs/1711.00937, 2017.

[68] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The Curious Case of Neural Text Degeneration. *CoRR*, abs/1904.09751, 2020.

[69] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *CoRR*, abs/2107.13586, 2021.

[70] Simon Willison. In Defense of Prompt Engineering. `https://simonwillison.net/2023/Feb/21/in-defense-of-prompt-engineering/`, February 2023.

Appendices

A   Benchmark Details

The full set of primitive tasks is summarized in Table 1 and the statistics detailed in Table 2. Primitives can be individually visualized in Figs. 7-16. Each primitive figure shows an analogy where the underlying relation is over a particular composable property of the scene/image.

Dataset	Primitive Task Set	Domain Size
3D Shapes	(1) Modify `object color` from source value to target value	10
	(2) Modify `floor color` from source value to target value	10
	(3) Modify `wall color` from source value to target value	10
	(4) Modify `scene orientation` from source value to target value	15
BitMoji Faces	(1) Modify `skintone` from source value to target value	3
	(2) Modify `hair style` from source value to target value	10
	(3) Modify `facial hair` from source value to target value	5
	(4) Modify `eyewear` from source value to target value	5
CLEVr Objects	(1) Add object to the scene at position p	1000
	(2) Delete object from the scene at position p	1000

Table 1: Summary of primitive tasks across all benchmarks

Dataset	Examples per task	#primitive tasks	#primitive extrapolation tasks	#k-composite tasks
3D Shapes	3	80000	1000	$1000 \ \forall k \in \{2, 3, 4\}$
BitMoji Faces	3	80000	1000	$1000 \ \forall k \in \{2, 3\}$
CLEVr Objects	2	55000	1000	$200 \ \forall k \in \{2, 3, 4\}$

Table 2: Benchmark statistics
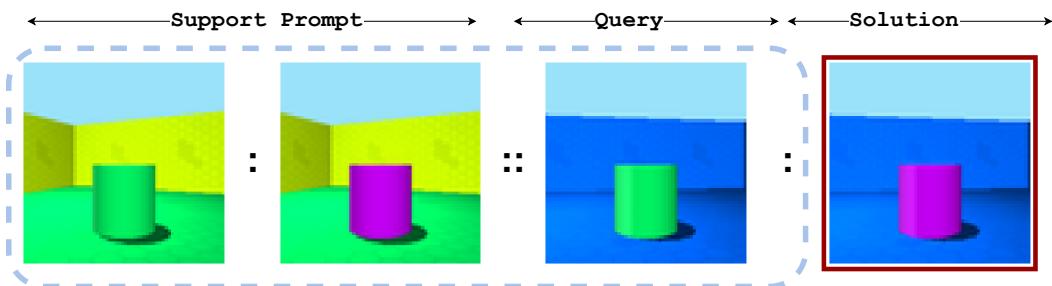
Figure 7: **3D Shapes** $\rightarrow$ Modify `object color` from source value to target value.

16
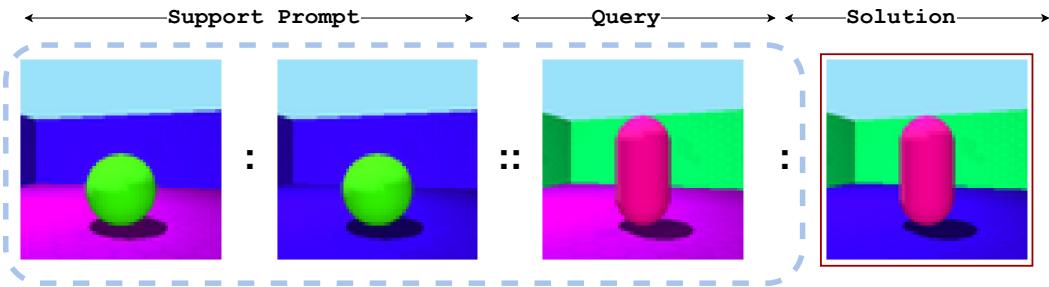
Figure 8: **3D Shapes** → Modify `floor color` from source value to target value.
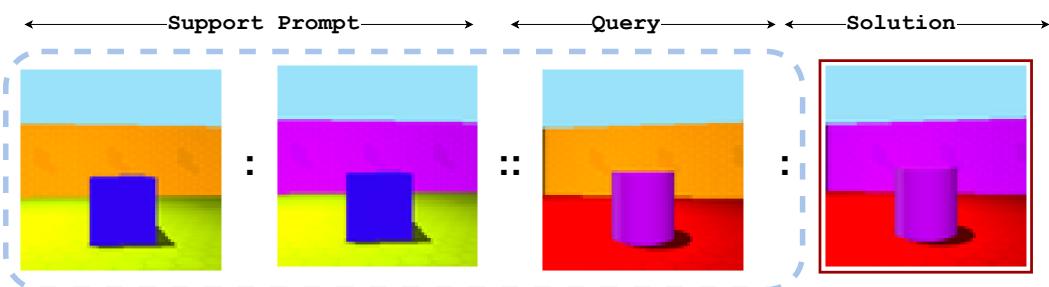
Figure 9: **3D Shapes** → Modify `wall color` from source value to target value.
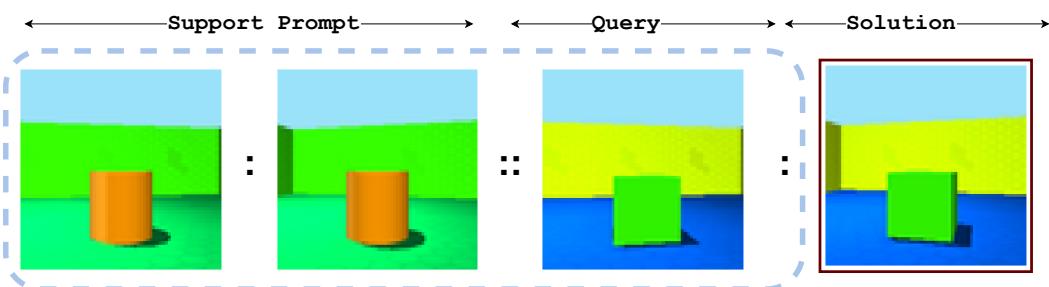
Figure 10: **3D Shapes** → Modify `orientation` from source value to target value.
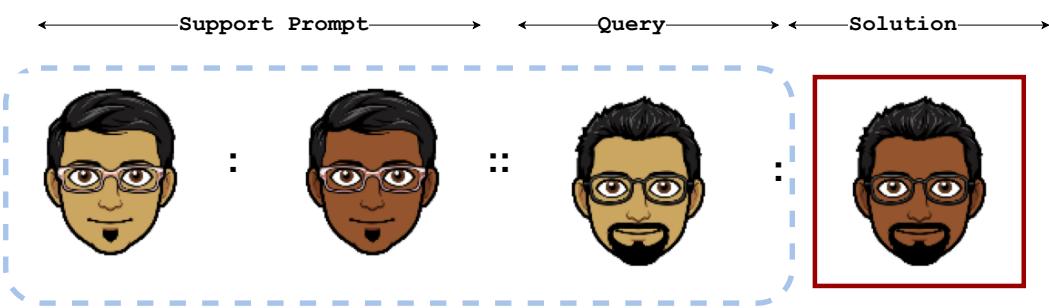
Figure 11: **BitMoji Faces** → Modify `facial skintone` from source value to target value.
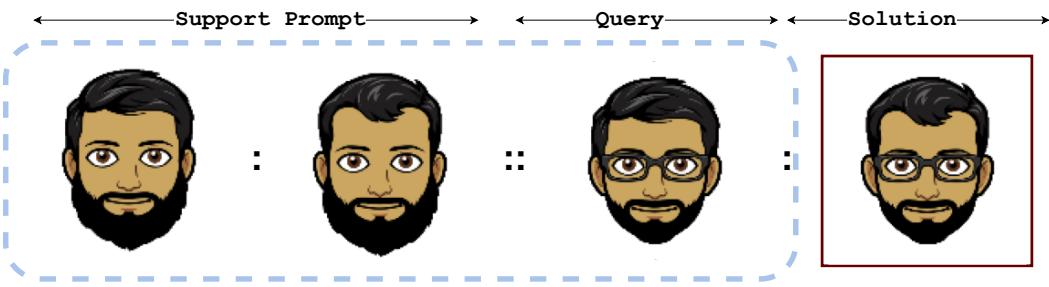
Figure 12: **BitMoji Faces** → Modify `hair style` from source value to target value.

17
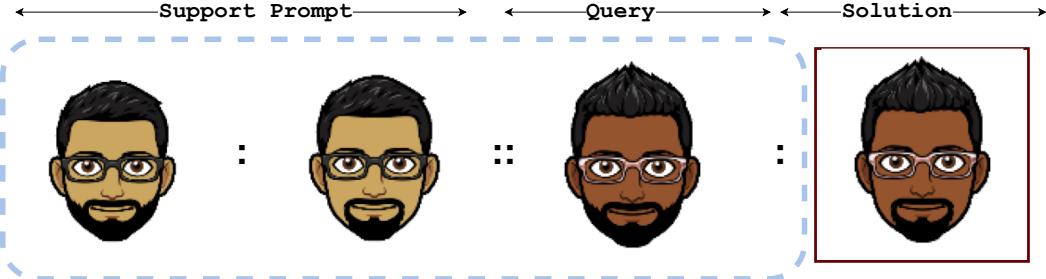
Figure 13: **BitMoji Faces** → Modify `facial hair` from source value to target value.
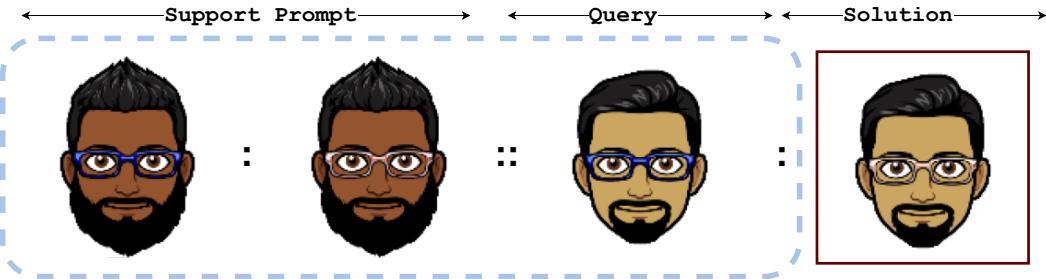
Figure 14: **BitMoji Faces** → Modify `eyewear` from source value to target value.
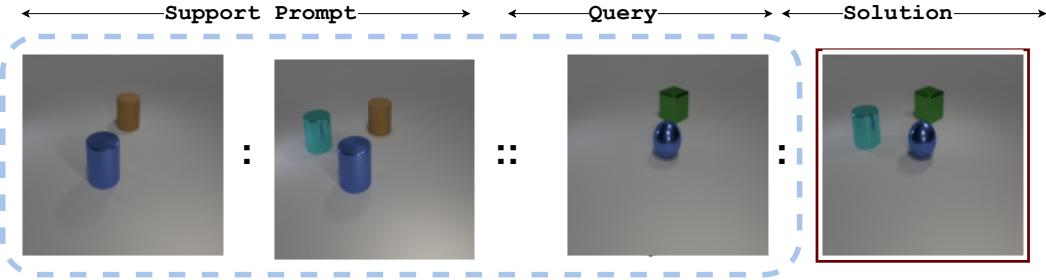
Figure 15: **CLEVr** → Add object to position p.
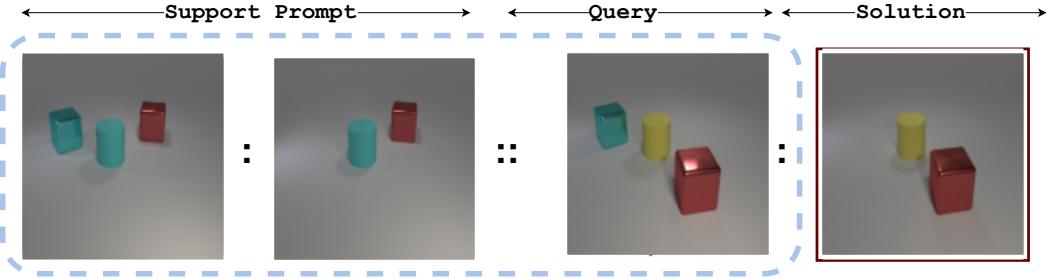
Figure 16: **CLEVr** → Delete object from position p.

B LLMs as In-Context Compositional Learners

It is straightforward to see how the aformentioned framework subsumes LLMs. The input space $\mathcal{X}$ is natural language and the compositional space $\mathcal{C}$ contains the dictionary of resultant tokens obtained via a tokenizer $\mathcal{E}_\phi(.)$. The relation class $\mathcal{R}$ in this context is the set of underlying tasks spanning numerical reasoning, reading comprehension, sentiment analysis, and question answering, to name a few. The core Transformer model forms the executor $\mathcal{T}_\alpha(.,.,.)$ that simply concatenates A, B, and C into a sequence prompt to compose the output D autoregressively. A de-tokenizer $\mathcal{D}_\theta(.)$ constructs natural language outputs from the generated tokens.

C Learning Agents

C.1 Monolithic: The encoder $e_\phi(.)$ and decoder $d_\theta(.)$ architectures are convolutional and deconvolutional networks with four layers initialized as shown in Tables 3 and 4, respectively.

Layer	Stride	Activation	Channels	Weight Matrix
Conv 3×3	2	ReLU	192	-
Conv 3×3	2	ReLU	192	-
Conv 3×3	2	ReLU	192	-
Conv 3×3	2	ReLU	192	-
Fully Connected	-	ReLU	-	$(192 \times \text{Image Size}/16 \times \text{Image Size}/16) \times 192$
Fully Connected	-	-	-	192×192

Table 3: Encoder $e_\phi(.)$ of the monolithic agent

Layer	Stride	Activation	Channels	Weight Matrix
Fully Connected	-	-	-	192×192
Fully Connected	-	ReLU	-	$192 \times (192 \times \text{Image Size}/16 \times \text{Image Size}/16)$
Conv Transpose 3×3	2	ReLU	192	-
Conv Transpose 3×3	2	ReLU	192	-
Conv Transpose 3×3	2	ReLU	192	-
Conv Transpose 3×3	2	ReLU	3	-

Table 4: Decoder $d_\theta(.)$ of the monolithic agent

The inference network $f_\alpha(.)$ and executor network $h_\beta(.)$ are feed-forward networks, as parameterized in Table 5.

Layer	Weight Matrix	Activation
Fully Connected	384×192	ReLU
Fully Connected	192×192	ReLU
Fully Connected	192×192	ReLU

Table 5: The inference network $f_\alpha(.)$ and executor network $h_\beta(.)$ of the monolithic agent

C.2 Object-Centric Learner: The Context Encoder Transformer (CET) is a stack of L blocks, each computing (a) self-attention over inputs followed by (b) cross-attention of inputs over context slots. The cross-attention implementation of block l is shown in Algorithm 2.

**C.3 Sequential Prompter:** We visualize this learner in Fig. 17.
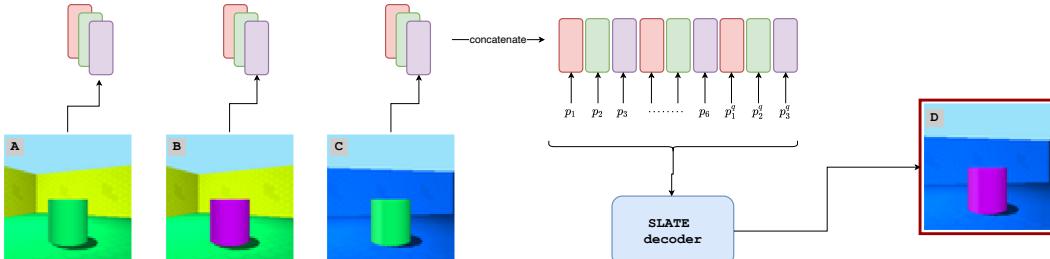
Figure 17: The Sequential Prompter concatenates slots from context prompts and a query together. It further injects position embeddings to create an LLM-like input prompt for the decoder.

Algorithm 2 Cross-Attention for Block l of CET

Require: $A^{1:N} \in \mathbb{R}^{N \times d}$, N slots of context A

Require: $B^{1:N} \in \mathbb{R}^{N \times d}$, N slots of context B

Require: $C_{SA,l}^{1:N} \in \mathbb{R}^{N \times d}$, N slots of input C obtained from the self-attention layer of block l

Output: $C_l^{1:N}$, Context encoded input slots of block l

Get query tokens: $Q^C = MLP_Q(C_{SA,l}^{1:N})$

Get keys, values of A, $K^A, V^A = MLP_{KV}(A_{l-1}^{1:N})$

Get keys, values of B, $K^B, V^B = MLP_{KV}(B_{l-1}^{1:N})$

Concatenate $K = [K^A, K^B] \in \mathbb{R}^{2N \times d}$
Concatenate $V = [V^A, V^B] \in \mathbb{R}^{2N \times d}$

Compute Cross-Attention Matrix $Att = \text{softmax}(Q^T K / \sqrt{d})$
Compute $C_l^{1:N} = Att \times V$

D Hyperparameters

The hyperparameters for OCL are listed in Table 6. The ablated architectures follow the same setup without the ablated module. All experiments were performed on NVIDIA A100 GPUs.

Module	Hyperparameters	3D Shapes	BitMoji Faces	CLEVr Objects
Pixel and Discrete Space	Image Size	64	128	128
	Image Tokens	256	1024	1024
dVAE	Vocabulary Size	512	1024	4096
	LR (no warmup)	5×10^{-5}	5×10^{-5}	1×10^{-5}
	Tau	0.1	0.1	0.1
CET	Num. Layers	4	4	4
	Heads	4	4	4
	Hidden Dims	192	192	192
Image GPT	Num. Layers	4	8	8
	Heads	4	8	8
	Hidden Dims	192	192	192
Slot Attention	Num. Slots	3	2	6
	Iterations	3	3	7
	Slot Heads	1	3	1
	Hidden Dims	192	192	192
	LR (no warmup)	5×10^{-5}	5×10^{-5}	1×10^{-5}
Training Setup	Batch Size	32	24	20
	LR Warmup steps	15000	30000	30000
	Peak LR	1×10^{-4}	1×10^{-4}	1×10^{-4}
	Dropout	0.1	0.1	0.1
	Gradient Clipping	1.0	1.0	1.0
Sampling Scheme	Temperature	0.7	0.7	0.1
	Top-k	8	8	−
	Top-p	0.75	0.75	0.5
Training Cost	GPU Usage	15 GB	40 GB	40 GB
	Days	1	3	5

Table 6: Hyperparameters for the Object-Centric Learner (OCL) instantiation and training setup

E Experiments

E.1 Object Slot Emergence: Distinct object slots stay preserved in object-centric architectures (OCL and Sequential Prompter) while training on primitive in-context learning tasks. We visualize the slot formation in Figs. 18, 19, and 20 for 3D Shapes, BitMoji Faces, and CLEVr Objects, respectively.


(a)   (b)
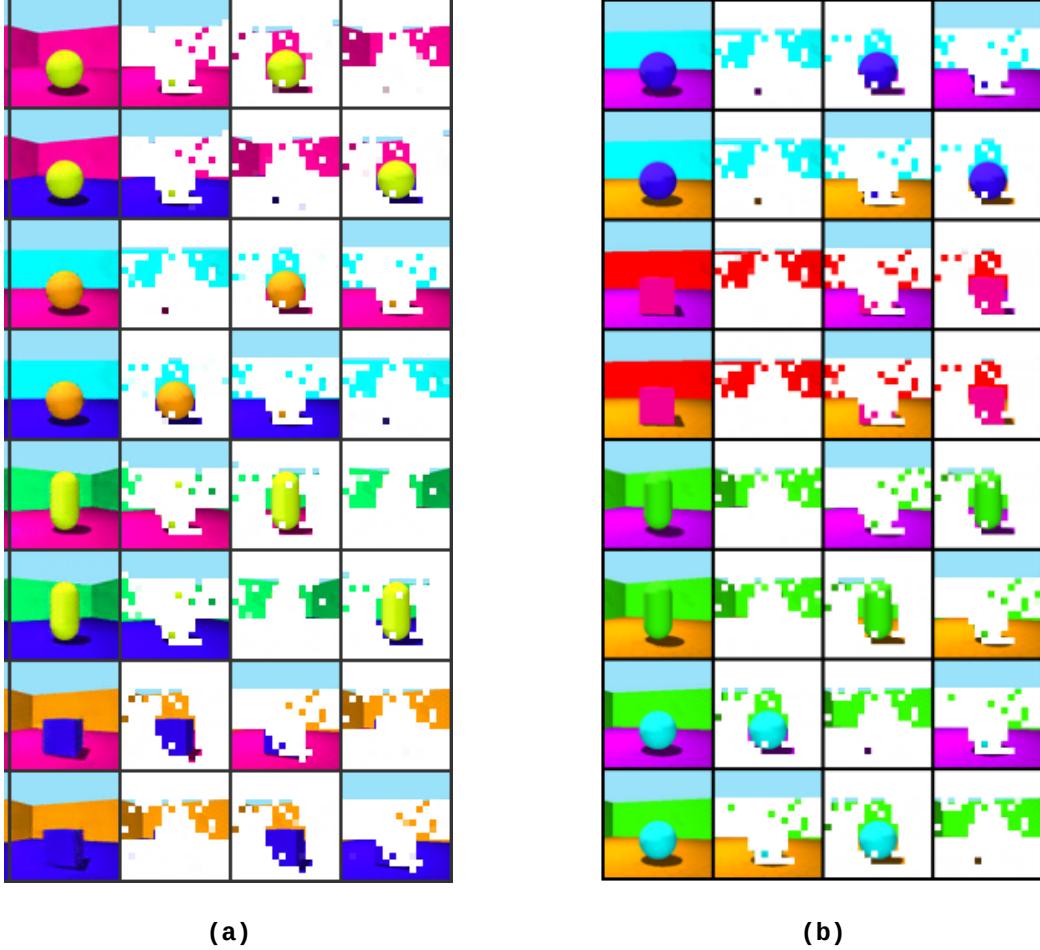
Figure 18: Slots from training OCLs on 3D Shapes: (a) pre-trained slots and (b) slots after Im-Promptu training. The structure of slots remains preserved.

E.2 Primitive Task Extrapolation: Figs. 21-23 show generated examples in the primitive task extrapolation regime by different agents.

E.3 Composite-Task Extrapolation: Figs. 24-30 show generated examples in the composite tasks extrapolation regime by different agents.

Broader Impact

This work demonstrates the potential for analogy-making to enable an in-context understanding of composition rules over visual stimuli. The benchmarks and meta-learning framework presented in this article provide a foundation for further exploration, which could lead to the development of more efficient and effective models for image generation and other visual tasks. While the current benchmark has no sampling bias over primitives, scaling up the models and the dataset can lead to biased outputs and the generation of counterfactual images capable of fooling humans.

(a)

(b)

Figure 19: Slots from training OCLs on BitMoji Faces: (a) pre-trained slots and (b) slots after Im-Promptu training. There occurs a minor modification in slots where the structure of the eye is merged with the slots carrying the facial contents.
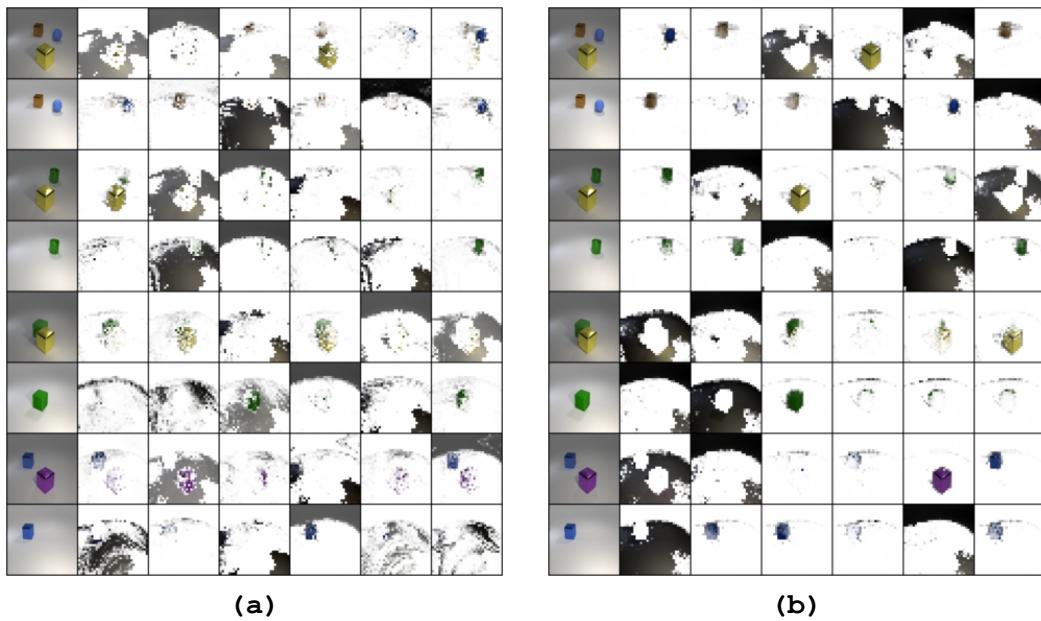
Figure 20: Slots from training OCLs on CLEVr Objects: (a) pre-trained slots and (b) slots after Im-Promptu training. We observe that slots become more refined after Im-Promptu training.
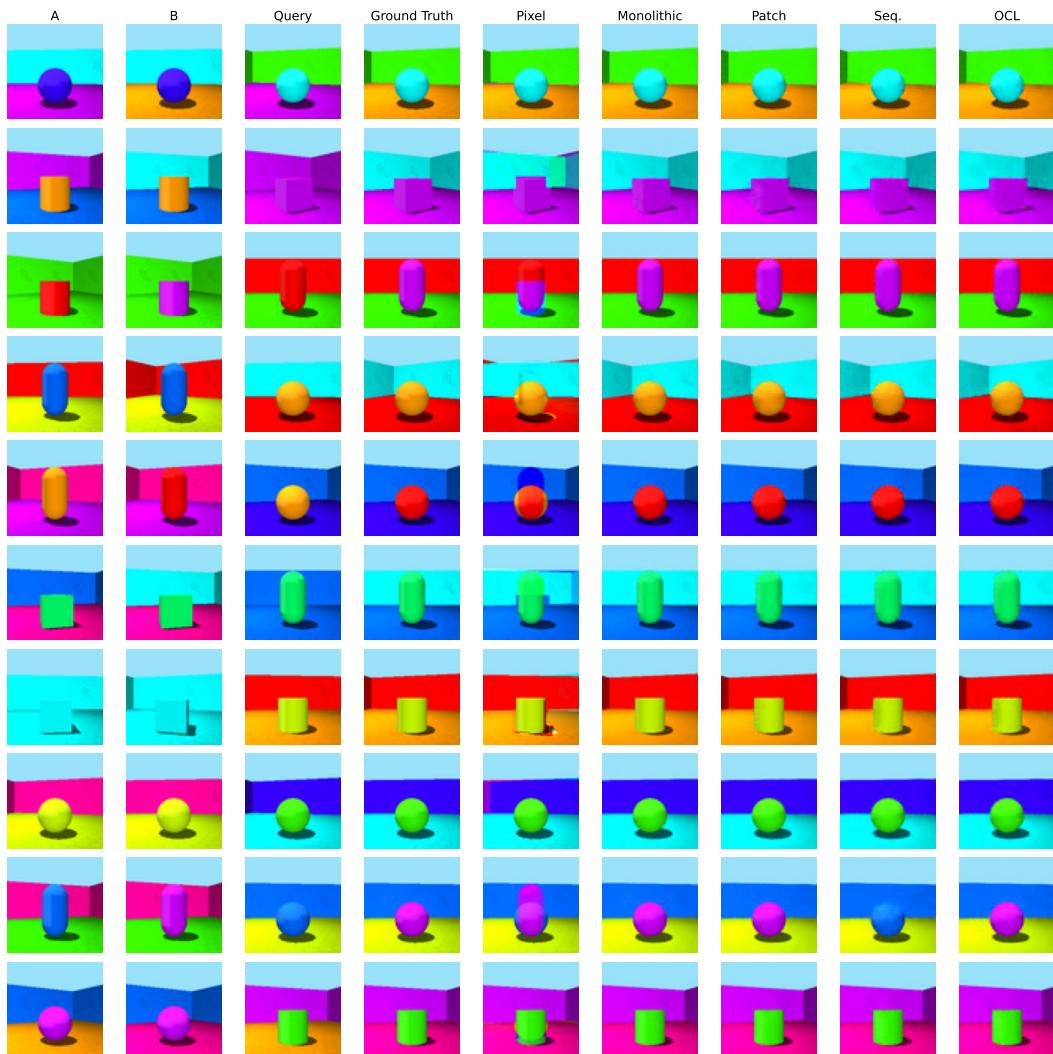
Figure 21: Primitive task extrapolation for the 3D Shapes benchmark.

Figure 22: Primitive task extrapolation for the BitMoji Faces benchmark.
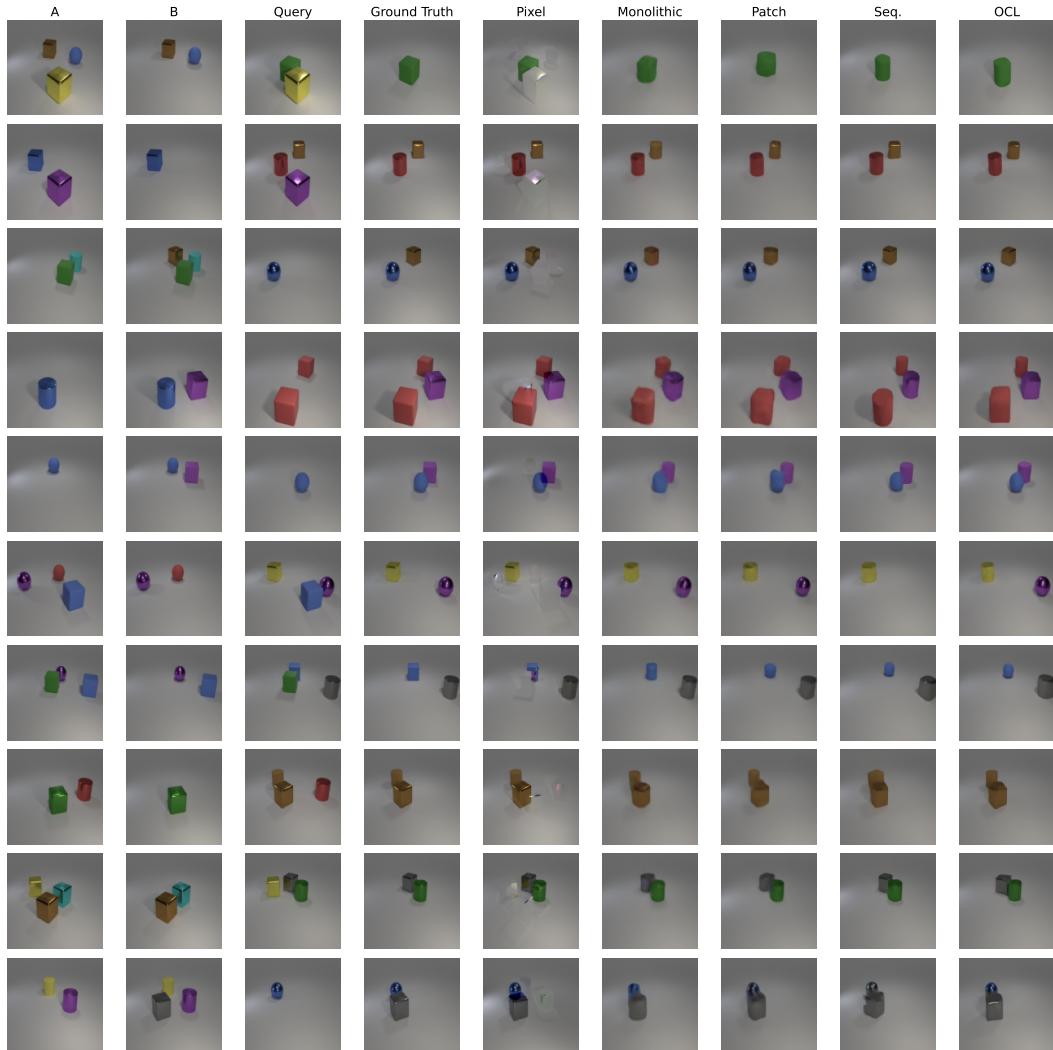
Figure 23: Primitive task extrapolation for the CLEVr Objects benchmark.
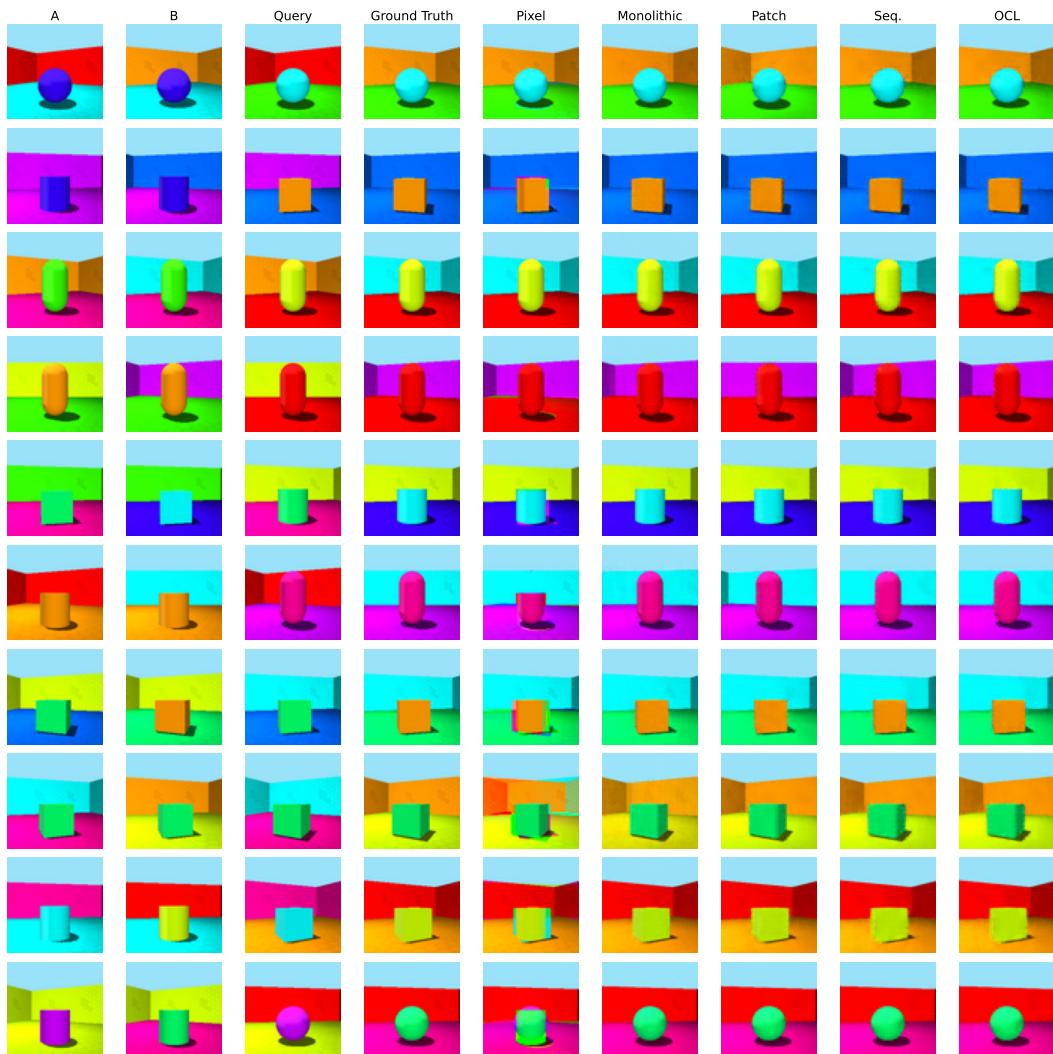
Figure 24: Two-composite task results on the 3D Shapes benchmark.
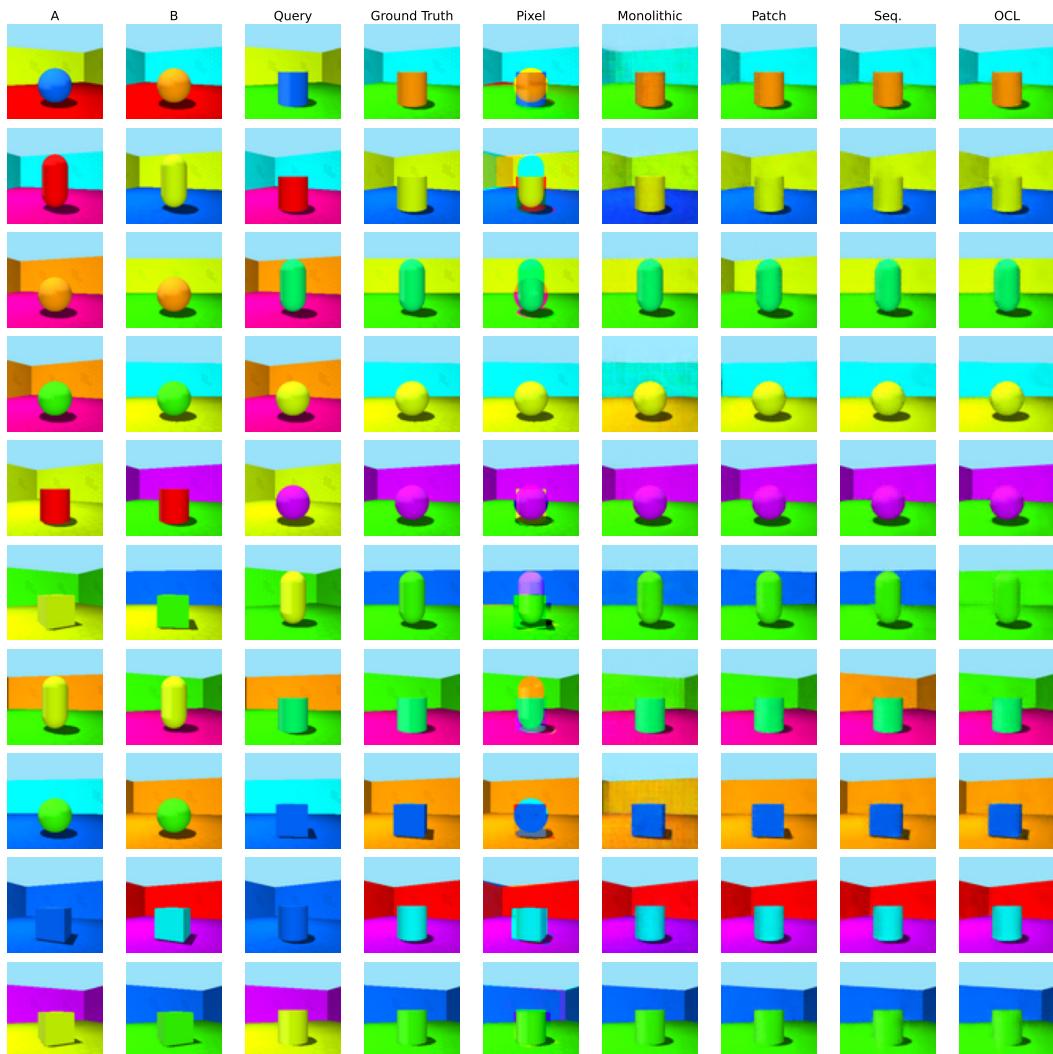
Figure 25: Three-composite task results on the 3D Shapes benchmark.
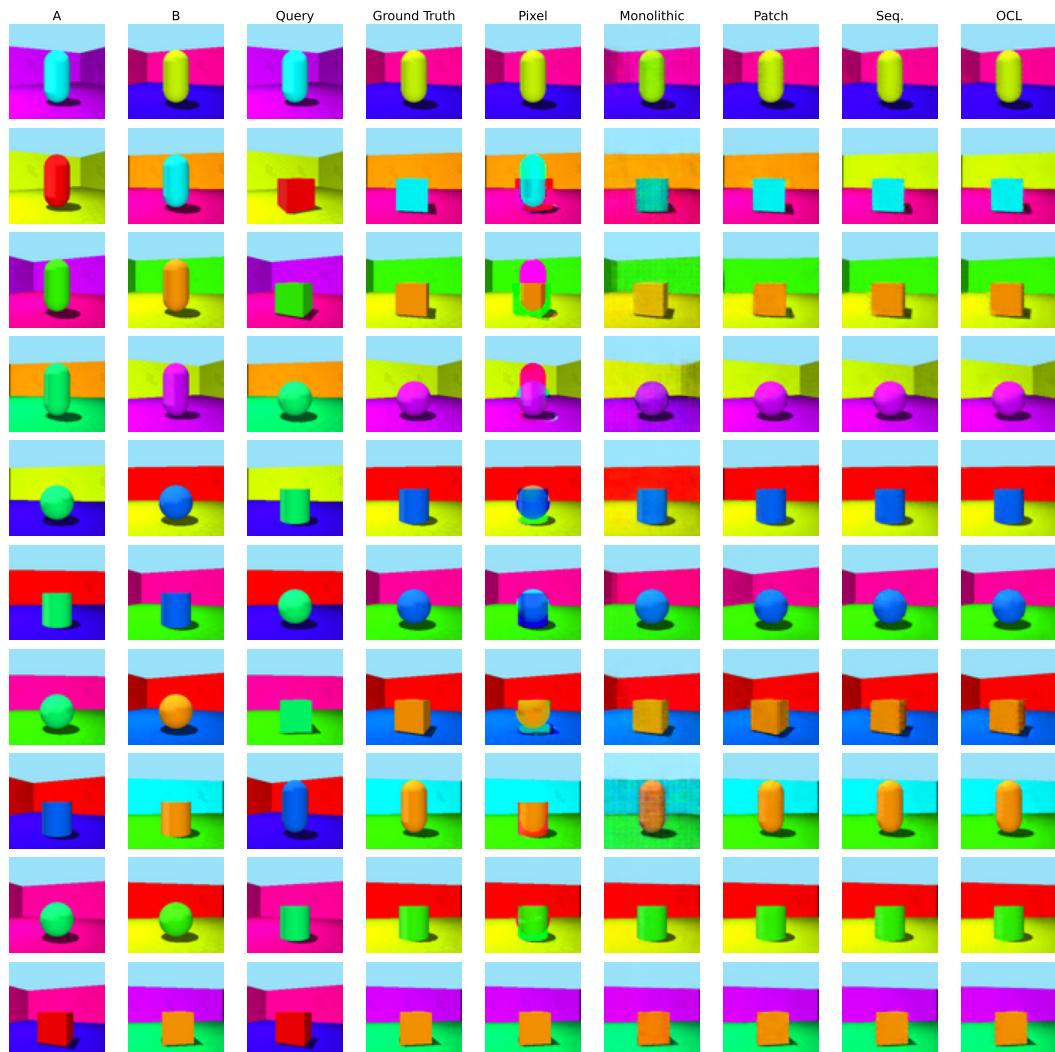
Figure 26: Four-composite task results on the 3D Shapes benchmark.
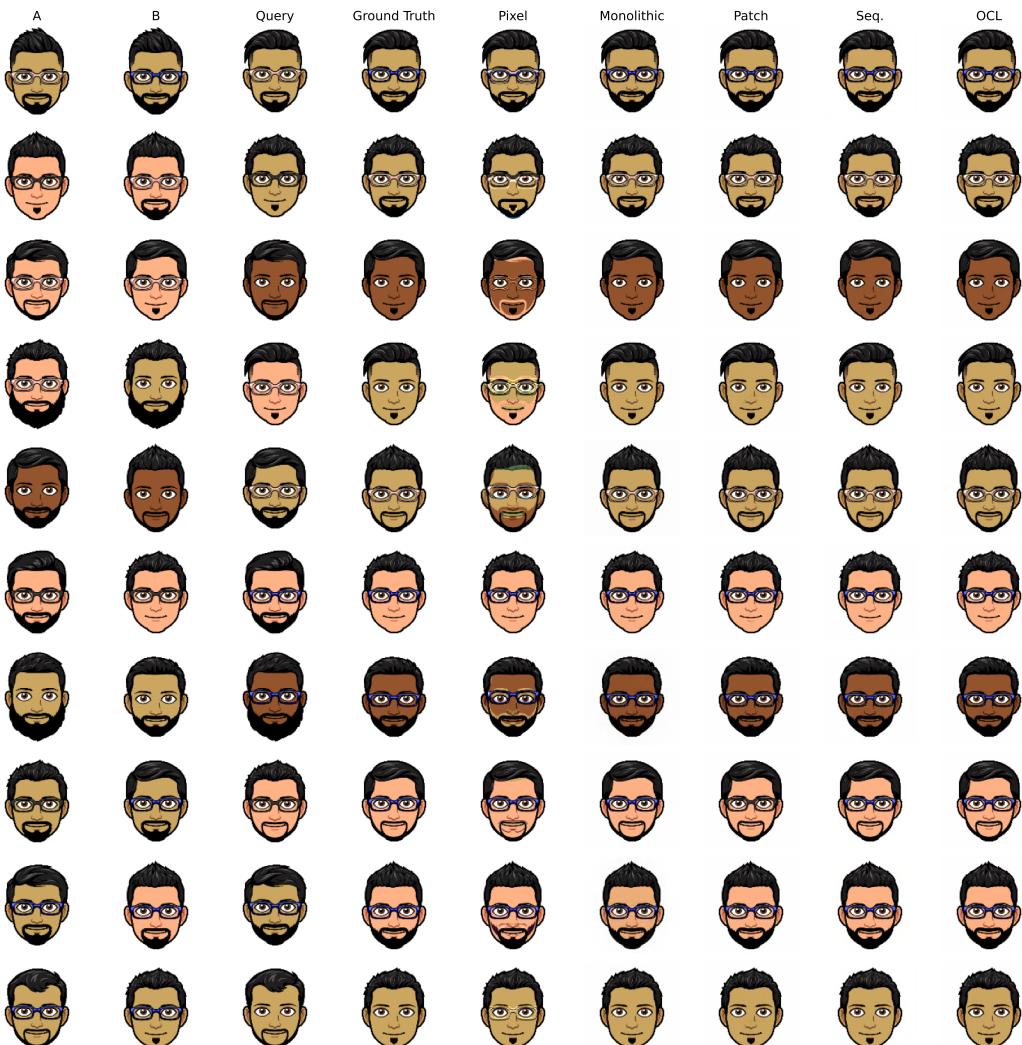
Figure 27: Two-composite task results on the BitMoji Faces benchmark.
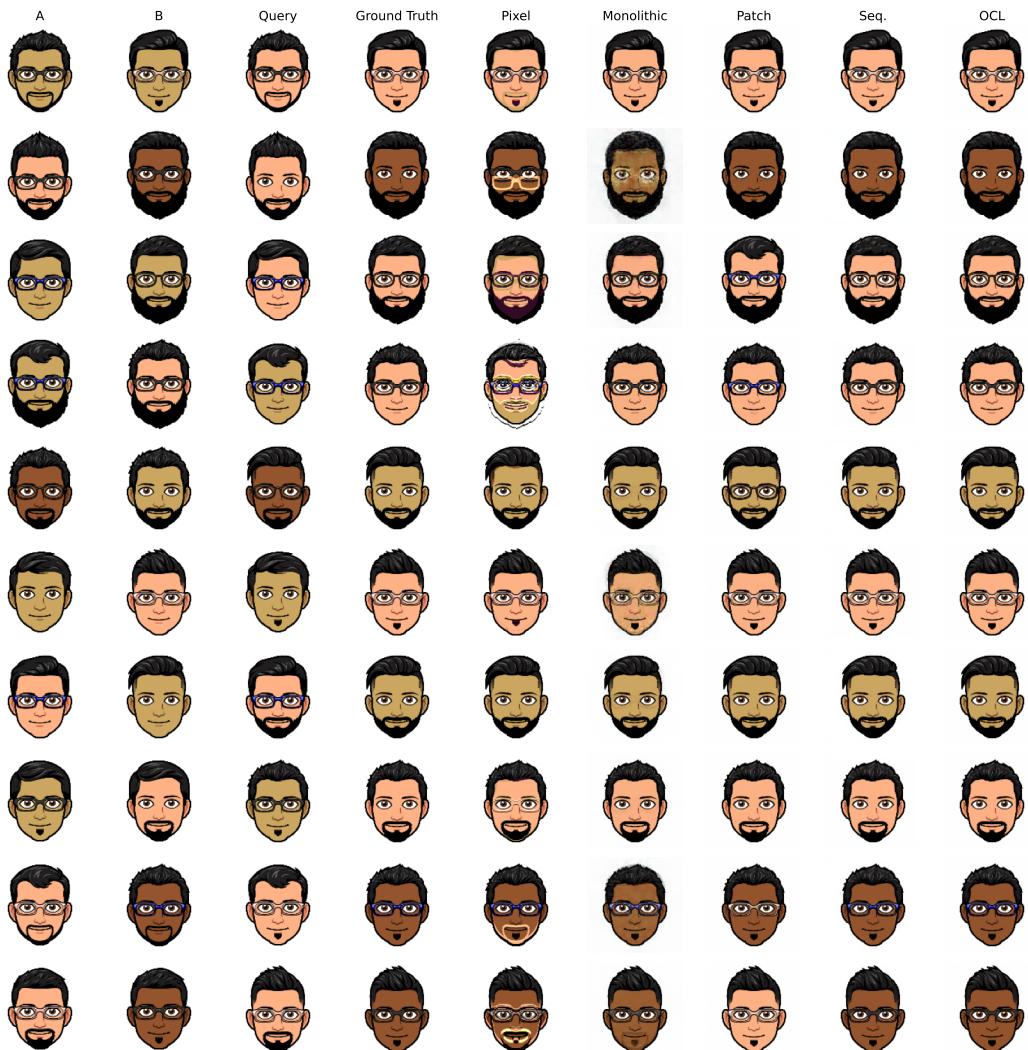
Figure 28: Three-composite task results on the BitMoji Faces benchmark.
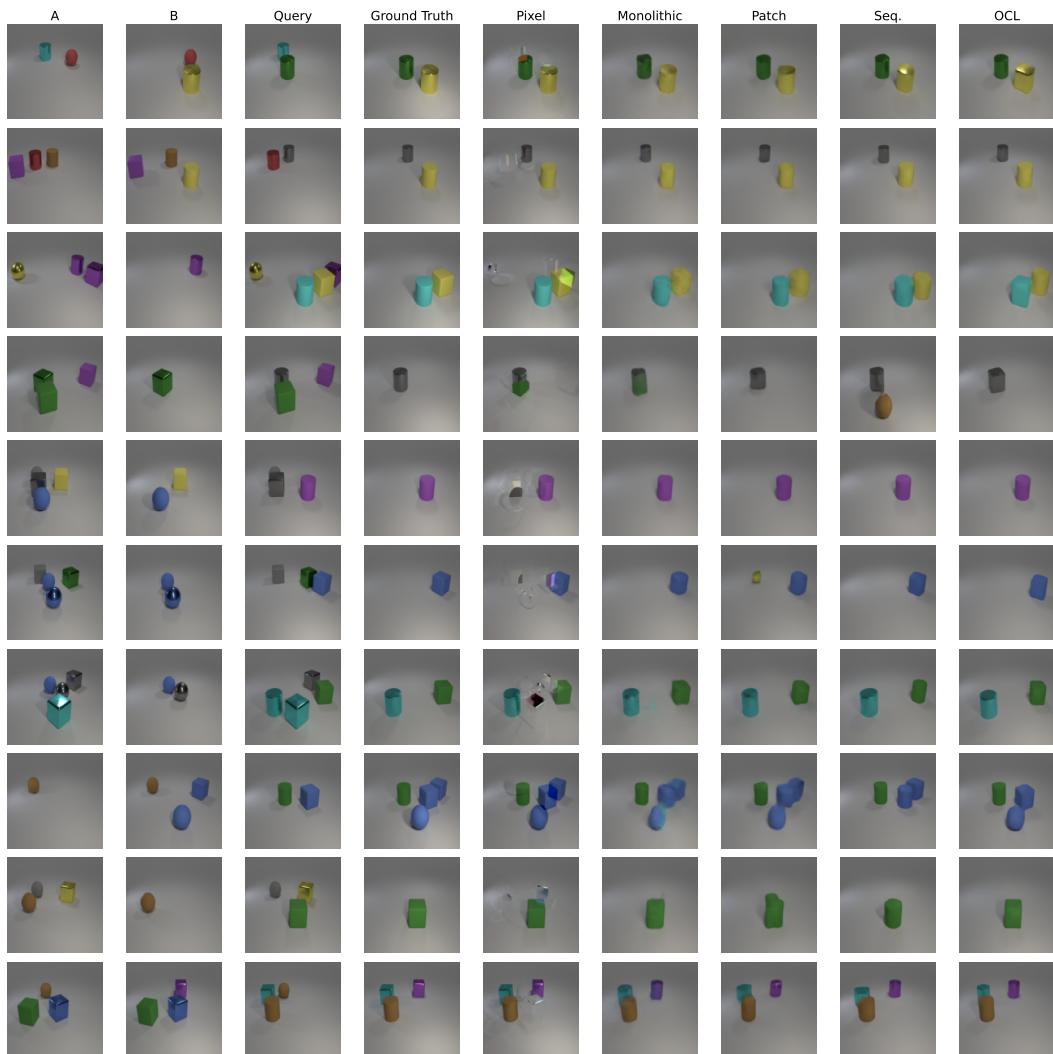
Figure 29: Two-composite task results on the CLEVr Objects benchmark.

Figure 30: Three-composite task results on the CLEVr Objects benchmark.