LINSLOT: EXPLOITING LINEAR REPRESENTATION HYPOTHESIS FOR UNSUPERVISED ATTRIBUTE DISCOVERY FROM SLOT BASED OBJECT REPRESENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper studies the problem of learning disentangled representations of objects and their attributes from raw, unstructured image data. Slot-based methods have shown considerable success in unsupervised learning of object representations from images. Block-slot attention-based methods extend this framework to attribute representations by assuming a uniform factorization of object representations into attributes, which may be suboptimal and consequently limit the quality of the learned representations. We therefore investigate a framework for jointly discovering object and attribute representations. Our key contribution is leveraging the Linear Representation Hypothesis (LRH), which postulates that composable concepts can be represented as linearly additive subspaces in slot representations. Based on this insight, we propose a probabilistic model connecting images, slots (objects), and blocks (attributes). We present an architecture that leverages block attention to connect attribute representations to slots and incorporates LRH in both object and attribute representation spaces. This architecture effectively optimizes the Evidence Lower Bound (ELBO) of the proposed graphical model. Our experiments demonstrate (i) effective discovery of disentangled object and attribute representations, (ii) empirical evidence for LRH in slot space, and (iii) the ability to perform image editing owing to the disentangled and interpretable nature of the learned representations. Our experiments on multiple datasets demonstrate improvements in DCI scores over state-of-the-art methods.

1 INTRODUCTION

A core aspect of the human intelligence is its ability to form disentangled representations of the world (Fodor & Pylyshyn, 1988; Whitehead, 1927; Dehaene et al., 2022), without having any success to supervised data. Enabling deep learning models with such representations has been a long-standing goal and is desirable for a number of reasons, i.e., having properties such as interpretability, compositional generalization, and sample efficiency (Bengio et al., 2013; Greff et al., 2020; Lake et al., 2017; Goyal & Bengio, 2022). In this work, we are specifically interested in the task of joint disentanglement of objects and their attributes (such as color, shape, size etc.) in static images without any supervision. Extracting such information automatically from images, in an unsupervised manner, will go a long way in building models, that have above desired properties of generalizability, sample efficiency and robustness to noise. Recently proposed *slot attention* based methods have shown promising results on learning *object-centric* disentangled representation in images(Locatello et al., 2020b; Akan & Yemez, 2025a; Didolkar et al., 2025), and these methods have been effectively applied for downstream tasks such as image segmentation, compositional generation and VQA. Extending this work, block-slot attention (Singh et al., 2023; Stammer et al., 2024) based methods partition the learned object representations (slots) into attributes (factors) (e.g., color, shape) yielding a fine grained and interpretable understanding of the image generation process.

Despite this progress, a fundamental question remains yet to be answered: is there a theoretical framework, in the form of a generative graphical model, which can guide the joint discovery of objects and their attributes in images in an unsupervised manner and provide a guarantee on the identifiability of latent factors under suitable assumptions? If so, what are these assumptions under which such idenfiability of factors (up to a permuation) holds, and how can we design an encoder-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

decoder architecture based on the assumptions of our model, to effectively discover the objects, and their latent factors from a given set of images.

Our solution to this problem is motivated by the linear representation hypothesis which has been used extensively for better understanding of latent factors in textual domains Park et al. (2024b). There have been some recent attempts to extend this work to the visual domains (Gao et al., 2025; Fel et al., 2026b), and that becomes our starting point. Specifically, we make an assumption that the object can be represented as a stochastic linear combination of its attribute values sampled from a given prior. Under this assumption, our graphical model can be described using a four step sampling process, going from sampling of a concept index (such as red , blue or green color), to sampling its specific value of the concept (degree of redness) from a prior distribution, expressing the object as a linear function of sample factor values, and then finally representing image as a learned function of its object representations. This graphical model enables derivation of an ELBO (evidence-lower-bound) formulation for optimizing the likelihood.

Correspondingly, our architecture, for disentanglement, involves a two-step encoder, going from an image to corresponding slots using existing slot-attention based methods (Locatello et al., 2020b) in the outer loop, which formulates this as a problem of competing attention over image patches (2) followed by block discovery module, which first projects the slot to a set of latent dimensions, over which blocks attend via competing attention, for the discovery of latent factors. The factors are poised to be linear combination of underlying (learnable) concepts, in line with earlier work (Singh et al., 2023), albeit with strong justification arising from our graphical model, for using the linear combination, as a way of imposing mixture of Gaussian prior. Decoding works in reverse, by first constructing the slots from the discovered factors, followed by construction of the image from the decoded slot. We experiment with two variants of our approach, one using a decoder working directly on the slot representation, and the other one, where the internal structure of each slot, in terms of its constituent blocks, is exposed to the decoder. Entire architecture is trained in an end to end manner, with a curriculum, for the joint discovery of objects (slots) and their attributes (factors).

We first show experimentally the validation of the linear structure in the slot space across the slots methods and datasets. We demonstrate the object manipulation via simple slot arithmetic of addition and subtraction. We perform extensive evaluation over three benchmark datasets to evaluate the efficacy of our approach both for object discovery as well as factor discovery. We find that our two stage pipeline not only results in improved factor discovery, it also improves the object discovery, compared to existing baselines, both in quantitative as well as qualitative evaluation. We also perform ablations demonstrating the contribution of various components our two-stage pipeline towards the efficacy of object and attribute discovery. We plan to release our code publicly on acceptance.

2 Related Work

Object-Centric Learning: Unsupervised object discovery methods aim at learning disentangled object representations for scenes without supervision. Earlier methods used generative modeling of the scene and imposed the object level structure in the latents (Eslami et al., 2016; Kosiorek et al., 2018; Jiang* et al., 2020; Lin et al., 2020; Kossen et al., 2020; Stanic & Schmidhuber, 2019; Stelzner et al., 2019). These *spatial attention methods* struggle to find occluded and non-local objects. *Scene mixture methods* models scenes as composition of object images Greff et al. (2017; 2019); Burgess et al. (2019); Engelcke et al. (2020; 2021) - however inference is computationally challenging. *Slot attention* Locatello et al. (2020b) leverages competitive spatial attention over the image to allow *slots* to bind to objects in an image. Others (Kipf et al., 2022; Elsayed et al., 2022; Wu et al., 2023a) model temporal dynamics and discover objects from images. Recent efforts extend the approach to realistic scenes by incorporating more expressive decoders Singh et al. (2022a;b); Jiang et al. (2023); Wu et al. (2023b) as well as using pre-trained image features (Caron et al., 2021) combined with a diffusion decoder Seitzer et al. (2023); Zadaianchuk et al. (2023); Singh et al. (2025); Akan & Yemez (2025b); Liu et al. (2025); Nguyen et al. (2026).

Disentangled representation learning: Attribute discovery aims at learning independent factored representations such as colour, shape, texture etc. that explain generation of objects observed in images. Earliest approaches train an VAE (Kingma & Welling, 2014) based methods (Higgins et al., 2017; Kim & Mnih, 2018; Chen et al., 2018) but suffer from identifiability issues (Locatello et al., 2018; Khemakhem et al., 2019) necessitating the need for inductive biases to aid disentanglement.

However with weak supervision it is guaranteed to recover attributes (Shu et al., 2020; Ahuja et al., 2022; Locatello et al., 2020a). Recently diffusion and transformer based methods were also proposed Yang et al. (2022; 2023). Enabling compositionality of representations also enabled disentanglement Jung et al. (2025).

Linear Representation Hypothesis: Presence of the linear structure in the textual domain well established (Mikolov et al., 2013; Park et al., 2024a; 2025; Jiang et al., 2024) even in LLMs. In vision domain Trager et al. (2023); Wang et al. (2023). However recovering such representations without supervision is difficult**?**. Existing work uses Sparse Autoencoders (SAEs) for recovering the *concepts* from latent representation which is additive in concepts(Gao et al., 2025; Fel et al., 2026b). However block have to identify all the values of the specific factor and SAE atoms fail to capture such representations (Bhalla et al., 2026). One option is to enable grouping of SAE atoms (Fel et al., 2026a) but it assumes the presence of the attributes is sparse which is not true for our proposed generative model.

3 Background: Slot Attention

Mathematically, given an input image $X \in \mathbb{R}^{H \times W \times 3}$ it is first passed through an encoder typically implemented as CNN to get the latent representation of the image, as a vector as $\mathbf{E} \in \mathbb{R}^{N_e \times d_{input}}$, where N_e represents the number of spatial features, and d_{input} represents their dimension. Following this, the each element in the slots set $\mathbf{S} \in \mathbb{R}^{N_s \times d_{slot}}$ is initialized by sampling from a normal distribution $\mathcal{N}(\mu, \Sigma)$ with learnable parameters. These slots then attend to the input image features with learnable k, q and v projection functions (matrices) to get the attention matrix A as:

$$A = \texttt{Softmax}\left(\frac{q(\mathbf{S})k(\mathbf{E})^T}{\sqrt{d_{slot}}}, \texttt{axis='slots'}\right) \tag{1}$$

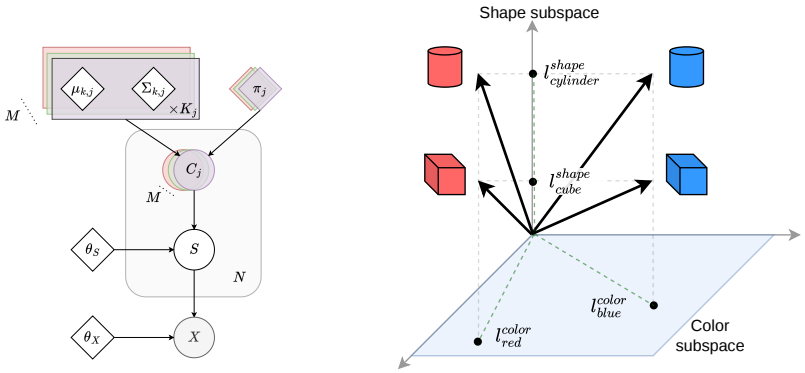
Additionally, the attention matrix A is further weight normalized along the 'axis' dimension to ensure competitive nature of the attention. The update vector $U = A * v(\mathbf{E}))$, resulting from the attention mechanism, is then fed to a recurrent neural network as input and previous slots as states. The output of the recurrent neural network is passed through an MLP to get the updated slot representation. This process is repeated for fixed number of iterations to get the final slot representation. Refer to Algorithm 1 in Locatello et al. (2020b) for a detailed pseudocode.

4 LinSlot

In this section, we present our framework LinSlot for unsupervised factor representation learning from *slots*. We first formulate a generative model of images that exploits the *linearity* of the slot representation space (Section 4.1). We then establish identifiability of the proposed generative formulation under weak assumptions (Section 4.2). Next, we describe an deterministic inference procedure motivated by this formulation, which first discovers object slots using Slot Attention and then infers attribute representations using our Block Attention mechanism (Section 4.3). Finally, we describe the training objectives motivated by the resulting ELBO and our curriculum-based training procedure (Section 4.4).

4.1 Generative model

In this section, we describe the generative model of the images considered in our work. We assume that each image consists of a set of objects, where each object is characterized by a set of factors of variation, such as color, shape, and size. Each factor is characterized by a fixed set of concepts, and each factor is associated with a particular concept from this set. For example, red, blue, and green are concepts for the color factor. The generative process proceeds in three steps: (1) for each factor of each object, a concept index is first sampled, after which the corresponding concept is sampled from its own prior and assigned to that factor. We assume the concept prior to be a normal distribution, which results in a GMM for the distribution of each factor; (2) the representation of each object is sampled conditioned on its factor values, and specifically, is given by the linear sum of the factor representations; and (3) the image is generated conditioned on the representations of all objects. Fig. 1a illustrates this generative process.



(a) Generative model. (b) Linear representation of slots.

Figure 1: Overview of the proposed representation and generative model.

Our problem statement then is: given a set of training images, a family of prior distribution p over concepts, an unknown function f mapping the sampled concept values to the corresponding object representation, and another function g mapping the set of object representations to the image representation, under what conditions, can we uniquely discover the (a) parameters of the concept prior (b) values of the underlying concepts upto a permutation, (c) inverse of the functions f, given fixed g? This leads to the following formulation and the corresponding identifiability result.

Specifically, assume that the image $X \in \mathbb{R}^{H \times W \times 3}$ is generated by set of N latent object representations referred as *slots* decoded by $\mathbf{S} = \{s_1, ..., s_N\}$. Each slot $s_i \in \mathbb{R}^{d_{\texttt{slot}}}$ itself is independently generated by M latent factor (attribute) representations referred as *blocks* denoted by the ordered set $\mathbf{c}_i = \{c_{1,i}, .., c_{m,i}\}$. Each jth block $c_{j,i} \in \mathbb{R}^{d_{\texttt{block}}}$ is sampled independently from it's own specific prior which is shared across all the objects. Let $\mathbf{C} = \{\mathbf{c}_1, .., \mathbf{c}_N\}$. The generative process can be expressed by following joint distribution:

$$p_\theta(X, \mathbf{S}, \mathbf{C}) = p_{\theta_X}(X|\mathbf{S}) \prod_{i=1}^{N} \left(p_{\theta_S}(s_i|\mathbf{c}_i) \prod_{j=1}^{M} p_{\theta_{C_j}}(c_{j,i}) \right) \tag{2}$$

Where $\theta = \{\theta_X, \theta_S, \{\theta_{C_j}\}_{j=1}^{M}\}$ are the parameters of the distribution. We further assume the prior on the blocks to be a Gaussian mixture model (GMM) which are specific to the blocks.

$$c_{j,i} \sim p_{\theta_{C_j}}(C_j) := \sum_{k=1}^{K_j} \pi_{k,j} \mathcal{N}(\mu_{k,j}, \Sigma_{k,j}) \tag{3}$$

Linear Structure of Slots: We also restrict the slot representations to be *linearly additive* in terms of some transformed block representations. That is

$$s_i \sim p_{\theta_S}(S|\mathbf{c}_i) := \mathcal{N}(\mu(\mathbf{c}_i), \Sigma_S) \quad where \quad \mu(\mathbf{c}_i) = \sum_{j=1}^{M} f_j(c_{j,i}) + b \tag{4}$$

Such additive structure implies the presence of the global *delta directions* in slots space which represents a direction from one factor value to another value. For example in Fig. 1b the delta direction of the blue to red color and cube to cylinder shape denoted as $\delta_{color}[red \rightarrow blue]$. These *delta directions* can be used to manipulate factors in the slot representation by simply vector addition. For example adding the delta direction $\delta_{color}[red \rightarrow blue]$ to any red object slot should change the object color to blue.

4.2 IDENTIFIABILITY

Informal Theorem 1 (Indentifiability of the Blocks). *Under the proposed generative model and the assumptions that* $\{f_j\}_{i=1}^{M}$ *are piecewise affine,* $Md_{block} \leq d_{slot}$ *and* $K_i \neq K_u K_w$ *for any*

triplets $i, u, w \in 1, .., m$, then the blocks are identifiable upto translation and permutation given slots are fixed.

The proof is provided in the A.1 which follows from the Kivva et al. (2022) where we first show that the concatenated blocks are also from GMM. Then the number of

4.3 INFERENCE

We approximate the true posterior $p_\theta(\mathbf{S}, \mathbf{C}|X)$ with variational approximation $q_\phi(\mathbf{S}, \mathbf{C}|X)$ due to intractibility of the former. We further factorize the variational posterior distribution as

$$q_\phi(\mathbf{S}, \mathbf{C}|X) = q_{\phi_S}(\mathbf{S}|X) \prod_{i=1}^{N} q_{\phi_C}(\mathbf{c}_i|s_i) \tag{5}$$

One benefit of having such inference modeling is that the slot posterior $q_\phi(\mathbf{S}|X)$ can be implemented as any slot based method and could also be pretrained. The block discovery method is not tied to the slot discovery as done in the block-slot attention which makes the optimization simpler. However inferring the blocks from slots is not trivial and we discuss it in following section.

The factorization in Eq. 5 suggests a sequential inference procedure: given an image X, we first infer the object slots $\mathbf{S}$ using a slot-based object discovery method, and then infer the attribute blocks $\mathbf{C}$ conditioned on the inferred slots. In our implementation, we use Slot Attention for the first stage and introduce Block Attention for the second stage:

$$X \xrightarrow{\text{Slot Attention}} \mathbf{S} \xrightarrow{\text{Block Attention}} \mathbf{C}. \tag{6}$$

The block inference module should therefore infer the blocks using the slots as input. Note that the slot representation is additive in terms of transformed block representations.

4.3.1 BLOCK INFERENCE USING BLOCK ATTENTION

We propose a *iterative refinement* based inference algorithm for inferring blocks from slots. Blocks are refined with the attention between projections of the slots along fixed learnable directions and the blocks inferred so far. The attention is normalized such that the blocks compete to attend for a particular projection of the slot.

Figure 2: Block Attention

1: Input: s_i
2: Initialize $\bar{\mathbf{c}}_i \sim \mathcal{N}(\mu_C, \Sigma_C)$
3: $c_{j,i} = \bar{c}_{j,i}\ \forall j \in \{1, \ldots, M\}$
4: $v_{k,i} = (s_i^\top u_k) u_k\ \forall k \in \{1, \ldots, N_p\}$
5: **for** $t = 1$ to T_b **do**
6: $\quad c_{j,i} = \texttt{LN}(c_{j,i})\ \forall j \in \{1, \ldots, M\}$
7: $\quad A_i = \texttt{Softmax}\left(\frac{k(\mathbf{V}_i) q(\mathbf{c}_i)^\top}{\sqrt{d_{\text{block}}}}\right)$
8: $\quad A_i = A_i / A_i.\texttt{sum}(\texttt{d='input'})$
9: $\quad U_i = A_i.v(\mathbf{e}_i)$
10: $\quad$ **for** $j \in \{1, \ldots, M\}$ **do**
11: $\quad\quad u_{j,i} = \texttt{GRU}_j(\texttt{st} = c_{j,i}, \texttt{in} = u_{j,i})$
12: $\quad\quad u_{j,i} = u_{j,i} + \texttt{MLP}_j(u_{j,i})$
13: $\quad\quad c_{j,i} = \texttt{Softmax}\left(\frac{u_{j,i} \mathcal{M}_j^\top}{\sqrt{d_{\text{block}}}}\right) \mathcal{M}_j$
14: $\quad$ **end for**
15: **end for**
16: **return** $\mathbf{c}_i$

Specifically, let $u_1, , ..u_{N_p}$ denote the N_p learnable vectors representing the latent space of projections where each $u_k \in \mathbb{R}^{d_{slot}}$. Let the projections of the slot s_i onto this space be denoted as $\mathbf{V}_i = [v_{1,i}, ..., v_{N_p,i}]$ where $v_{k,i} = (s_i^T u_k) u_k$. Given these projection vectors represented in the latent space, we infer the blocks corresponding to a slot s_i by iteratively refining initial block set $\bar{\mathbf{c}}_\mathbf{i}$ with the attention on to these projection vectors. The attention is made competitive by normalizing along block axis (Line 8, Algorithm 2). In line 9, we normalize the attention matrix again along input (project vector) axis, normalization across 'input' (projection vectors) dimension. U_i (line 11) denotes the update vector computed as a result of the attention mechanism (line 11).

The updates corresponding to particular blocks $c_{j,i}$ denotes as $u_{j,i} \in \mathbb{R}^{d_{block}}$ are used to refine the blocks from previous iterations with the GRU and MLP shown in the algorithm. Finally at the end of each refinement step, the blocks are also replaced as linear sum of the prototypes similar to block slot attention Singh et al. (2023) (line 15).

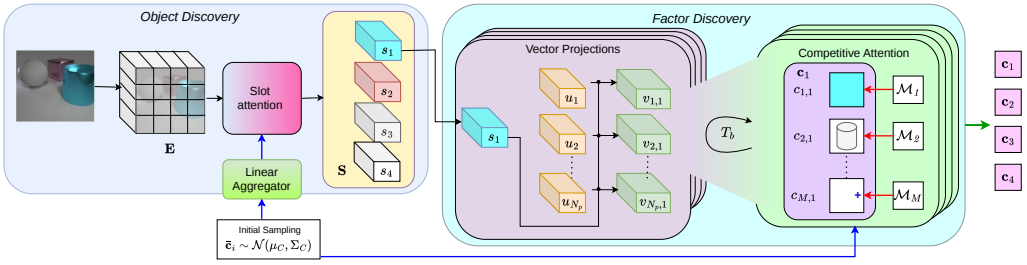


Figure 3: **Overview of the inference:** The image is encoded into spatial features $\mathbf{E}$, from which slots $\mathbf{S}$ are inferred via slot attention. For each slot, projected features $\mathbf{v}$ are used to iteratively refine M attribute blocks through competitive attention, with each block constrained to a linear combination of shared concept vectors. The resulting blocks serve as disentangled attribute representations.

4.4 Learning and Implementation

The parameters θ and ϕ can be learned by maximizing the evidence lower bound (ELBO) of the data log-likelihood. For the proposed generative model, the ELBO is given by: For the full derivation, please refer to Appendix A.2.

$$\log p_\theta(X) \geq \underbrace{\mathbb{E}_{q_{\phi_S}(\mathbf{S}|X)}\left[\log p_{\theta_X}(X|\mathbf{S})\right]}_{\text{Reconstruction}} - \underbrace{D_{KL}(q_{\phi_S}(\mathbf{S}|X)\|p_{\theta_S}(\mathbf{S}|\mathbf{C}))}_{\text{Slot regularizer}} \\ \underbrace{- \sum_{i=1}^{N} \mathbb{E}_{q_{\phi_S}(\mathbf{S}|X)}\left[D_{KL}(q_{\phi_C}(\mathbf{c}_i|s_i)\|p_{\theta_C}(\mathbf{c}_i))\right]}_{\text{Block regularizer}}. \tag{7}$$

The first term corresponds to the reconstruction objective, which measures the reconstruction error between the input images and the images reconstructed from the inferred slots. The second term is the slot regularizer, which encourages the inferred slots to be close to the linear combination of the transformed blocks. The third term is the block regularizer, which penalizes deviations of the inferred block distributions from their GMM priors.

In our implementation, we optimize the image reconstruction loss and the slot regularization loss explicitly, while the block regularizer is enforced implicitly through the prototype memory. Furthermore, when the mixture assignment probabilities are inferred using cosine similarity, the block projection in the prototype space (Line 13, Algorithm 2) is equivalent to explicitly setting the block regularization loss to zero. See Appendix A.3 for the proof. Thus, we optimize the following loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{img-recon}} + \lambda \mathcal{L}_{\text{slot-reg}}. \tag{8}$$

We implement the slot-specific inference network q_{ϕ_S} using Slot Attention and the block-specific inference network q_{ϕ_C} using Block Attention (Section 4.3.1). The first loss is the image reconstruction loss. Depending on the decoder architecture, this loss can be implemented as an MSE loss for CNN-based decoders, a dVAE prediction loss for transformer-based decoders, a denoising loss for diffusion-based decoders, or a feature reconstruction loss. The second loss is the slot regularization loss, defined as $\mathcal{L}_{\text{slot-reg}} = \sum_{i=1}^{N} \|s_i - \hat{s}_i\|$ where s_i denotes the slots inferred by Slot Attention and $\hat{s}_i$ denotes the slots generated from the inferred blocks, given by $\hat{s}_i = \sum_{j=1}^{m} f_j(c_{j,i}) + b$.

We train the model in two phases. In the first phase, we train only the Slot Attention module using the image reconstruction loss. In the second phase, we introduce the block-specific modules and train the full model using the losses described above. For the slot-specific reconstruction loss, we found it useful to also use slots generated from the inferred blocks. Specifically, in each mini-batch, we use half of the slots inferred by Slot Attention and half of the slots generated from the inferred blocks. In 2nd phase, instead of sampling the initial slots from a normal distribution, we initialize them using linear combination of the transformed initial blocks. Specifically, the slot s_i in Slot Attention is initialized as $s_i = \sum_{j=1}^{m} f_j(\bar{c}_{j,i})$. Line 2 of Alg. 2 for block initialization procedure.

Table 1: Cosine similarity between two estimations of the same delta direction.

	CLEVR Easy		CLEVR Hard		CLEVR-Tex		MOVi-C
Method	**Slot Attn.**	**SLATE+**	**Slot Attn.**	**SLATE+**	**SLATE+**	**Slot Diff.**	**CODA**
Same δ	0.6610	0.7465	0.6024	0.6511	0.5618	0.6356	0.2230
Different δ	0.2232	0.1647	0.1898	0.1493	0.1498	0.1719	0.1020

5 EXPERIMENTS

Our experiments try to answer the following questions: (1) Are slot representations linear in terms of the object factors? (2) Is the proposed attention-based inference method capable of recovering objects and factors of objects? (3) How much is the model robust to its training pipeline and its architecture decisions?

5.1 SETUP

Methods: We evaluated various slot based methods for the validating the linearity in their representations. This includes the 1. Slot attention Locatello et al. (2020b) which uses SBD decoder Watters et al. (2019). 2. SLATE Singh et al. (2022a) which uses the transformer based decoder. We found that modified the SLATE with difference slot initialization mechanism and a small module before decoder improves the SLATE performance. Please refer to Appendix C.1. We name this baseline as SLATE+. 3. Slot-Diffusion Wu et al. (2023b) which uses the diffusion decoder. 4. DINOSAUR-FT Didolkar et al. (2024) which uses feature reconstruction loss 5. CODA-base Nguyen et al. (2026) which uses the denoising diffusion loss from pretrained diffusion. We do not include the contrastive alignment loss form the original work.

For evaluating the unsupervised object and attribute discovery we use Slot attention, SLATE+ and a block-slot attention based baseline Sysbinder Singh et al. (2023). For our proposed attention based method LinSlot we use SLATE+ as slot base which we train in the first phase. To see the effect of the attention based inference method alone we also include our attention based sequential inference method with *block decoder* instead of slot decoder. We name it as LinSlotB. LinSlotB is trained end to end without slot regularizer.

Datasets: We use three image benchmarks derived from Johnson et al. (2016); Karazija et al. (2021), provided by Singh et al. (2023) for unsupervised object and attribute discovery. These datasets consist of the multi object images with 128×128 resolution. The objects in these datasets are characterized by set of categorical and continuous attributes. CLEVR-Easy, CLEVR-Hard and CLEVR-Tex. Additionally for slot linearity validation we further use MOVi-C Greff et al. (2022) which The frames from video as used as images. We generate the paired version of the datasets where each datapoint is 4 images. The first two images are same except one object is different which have a single attribute different . The remaining two images are also same have all objects same except one object which are different in single attribute same as the attribute in first pair and the values of the changed attribute matches that from the first pair. For further details please refer to B.2

Metrics: We evaluate the linearity of slot representation by cosine similarity between the two estimations of the *delta directions* 4.1. That is subtraction of slot pair differing in only one attribute should be similar to the subtraction of different slot pair differing in same one attribute with similar values. We evaluate the attribute disentanglement quality of representation using the standard DCI metric Eastwood & Williams (2018), which measures Disentanglement, Completeness, and Informativeness. Additionally, we report the Foreground Adjusted Rand Index (FG-ARI) to quantify object discovery. FG-ARI measures the clustering agreement between predicted slots and ground-truth masks exclusively on foreground objects, providing as assessment of segmentation quality.

5.2 LINEARITY IN SLOTS

As mentioned above, the presence of linear structure in the slot space implies the existence of *global delta directions* 4.1. We test this by measuring the cosine similarity between two estimates of the same delta direction obtained from different object slots in two different images. For example, for the color factor, we compare $\delta_{\text{color}}[\text{red} \rightarrow \text{blue}]$ estimated from two different object slots belonging

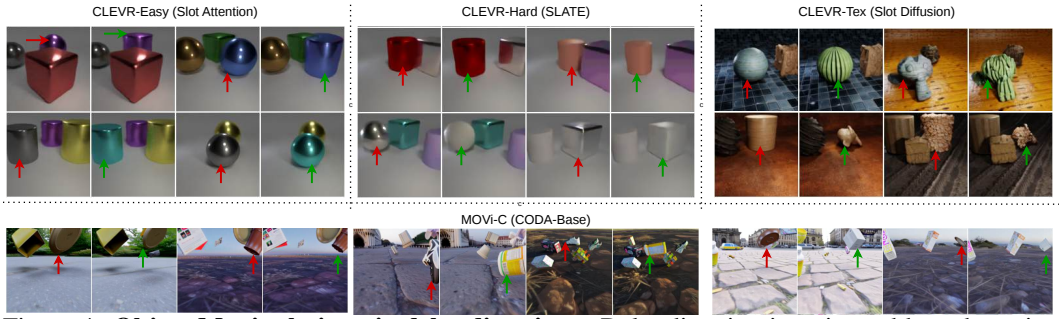


Figure 4: **Object Manipulation via delta directions:** Delta direction is estimated by subtracting slot of object indicated from first image from slot of the object indicated in second image. This delta direction is then added to the object indicated in the third image. The fourth image is decoded by slots manipulated slots from third image. First three images are reconstruction.

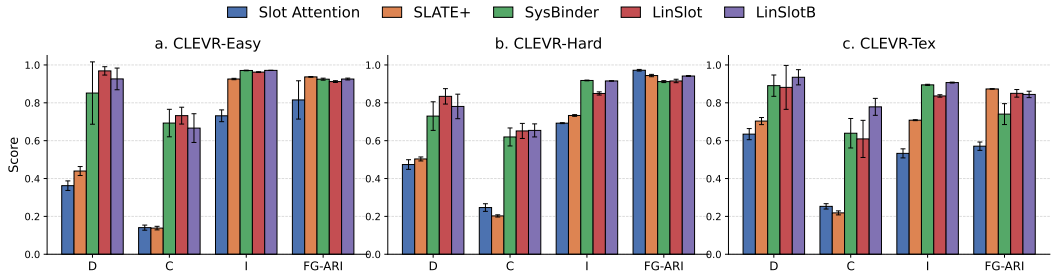


Figure 5: **Quantitative Results:** Comparison of DCI and FG-ARI scores for our model and baselines across all datasets.

to two different images. As a baseline, we measure the cosine similarity between delta directions corresponding to different factors. Table 1 shows the results across four datasets and four different methods. The cosine similarity between two estimates of the same delta direction (first row) is consistently higher than that between delta directions from different factors. This trend holds across all slot methods, including those using pretrained diffusion decoders, and across all datasets. These results suggest the existence of global delta directions and, consequently, linearity in the slot space, which is inherent to slot attention rather than specific to a particular dataset or model. On MOVi-C, CODA exhibits lower absolute values, which we attribute to errors in slot inference due to over- or under-segmentation.

We also manipulate slots using these delta directions, analogous to word-embedding arithmetic, where $r(\text{king}) - r(\text{boy}) + r(\text{girl}) \sim r(\text{queen})$. Specifically, we modify a slot by adding a delta direction and decode it using the slot decoder to generate the manipulated image. Figure 4 shows the results. Changing a slot by adding a delta direction consistently changes the corresponding object factor in a predictable manner. This behavior holds from the weaker SBD slot decoder to the strongest diffusion-based decoder, suggesting that linearity is fundamental to slot representations.

5.3 Object and Factor Discovery

Fig. 5 shows the DCI and FG-ARI scores for all the methods across all the datasets. Our attention-based methods with the slot decoder, LinSlot, and the block decoder, LinSlotB, perform better on the CLEVR-Easy and CLEVR-Hard datasets than all the baselines on the D, C, and FG-ARI scores. On CLEVR-Tex, LinSlot is on par with the best method on the D and C scores, whereas its FG-ARI score is comparable to the best number, which is achieved by LinSlotB. This highlights the effectiveness of our approach, which does not require specialized decoders or the extensive hyperparameter tuning required by block-slot attention. The I score is lower than the best-performing method on the CLEVR-Hard and CLEVR-Tex datasets. However, when changing the decoder to the block decoder, as in LinSlotB, the I scores match the best-performing values, suggesting the role of the decoder in the I score. The variance of both our methods is lower than that of the other baselines, suggesting lower sensitivity to the random seed. This suggests that our identifiability analysis is effective.

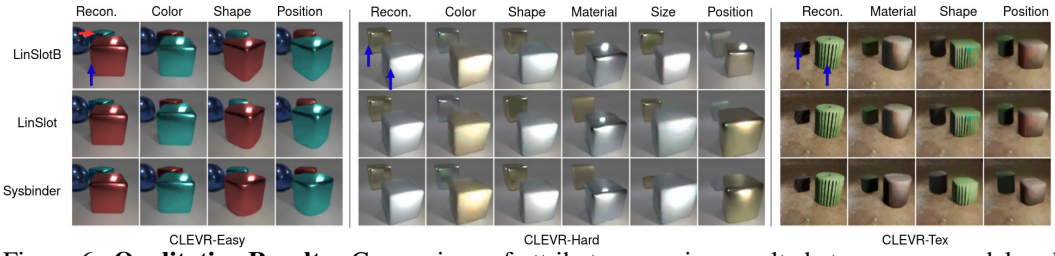


Figure 6: **Qualitative Results:** Comparison of attribute swapping results between our model and SysBinder. Blue arrows indicate the objects participating in the swaps. *Recon.* denotes the image reconstructed by the model.

Table 2: Effect of Training Choices

2-PHASE	BOTH	D	C	I	FG-ARI
✓	✓	0.9686	0.7322	0.9621	0.9122
×	✓	0.8466	0.5994	0.9487	0.9055
✓	×	—	—	—	—
×	×	0.7592	0.6104	0.9368	0.9307

Table 3: GRU-MLP Sensitivity

GRU	MLP	D	C	I	FG-ARI
✓	✓	0.9686	0.7322	0.9621	0.9122
×	✓	0.9807	0.7322	0.9634	0.9113
✓	×	0.9710	0.7356	0.9619	0.9158
×	×	0.9767	0.7468	0.9656	0.9189

Additionally, we perform a swapping experiment in which we swap the blocks between two objects from the same image and decode the swapped representations to see whether the factors change in the intended way. Fig. 6 shows the results of both our methods along with SysBinder on all three datasets using their best seeds. We observe that on CLEVR-Easy, all methods perform the swapping faithfully. On CLEVR-Hard, we observe that even the best seed of SysBinder struggles with the size factor. On CLEVR-Tex, however, we observe that all the methods perform the swapping faithfully.

5.4 ABLATIONS

Effect of Training Choices: Our method, LinSlot, is trained in two phases and uses both the slots generated from the inferred blocks and the slots inferred directly from the images for decoding. We ablate the impact of these training choices, as they are not trivial consequences of the generative model. The results are shown in Table 2. Removing either choice substantially affects the D and C scores, whereas removing the two-phase training and using slots inferred directly from the images leads to training collapse.

GRU-MLP Sensitivity: We use a GRU and MLP combination in the refinement loop of the blocks. This choice was inspired by Block-Slot Attention Singh et al. (2023). We ablate these design choices; the results are shown in Table 3. We note that LinSlot is not sensitive to the choice of GRU and MLP, unlike Block-Slot Attention, highlighting robustness of our model.

6 CONCLUSION AND LIMITATION

In this work, we presented a generative framework for unsupervised object and factor discovery that exploits the linear additive structure of factors in slot representations. To the best of our knowledge, we are the first to demonstrate the Linear Representation Hypothesis (LRH) in slot space. We further proposed a sequential inference framework and derived its learning objectives from ELBO optimization. Experiments across datasets and slot-based methods provide consistent evidence for linear structure in slot representations, while our attention-based block inference effectively discovers both objects and their factors. Our model does not explicitly represent background or global properties such as lighting and style, which are treated as sources of uncertainty. Future work includes end-to-end identifiability, extension to video, and explicit modeling of global properties and object interactions.

7 AI USE STATEMENT

In this work, we used generative AI tools for grammar and punctuation refining of text. We have not used generative AI tools for creating ideas, writing core code of the model, writing proofs. Additionally, we used generative AI tools for creating plot code. We have reviewed all AI-assisted work. We verified the grammar and graph generated. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Kartik Ahuja, Jason Hartford, and Yoshua Bengio. Weakly supervised representation learning with sparse perturbations. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022.

Adil Kaan Akan and Yücel Yemez. Slot-guided adaptation of pre-trained diffusion models for object-centric learning and compositional generation. In *International Conference on Learning Representations*, 2025a.

Kaan Akan and Yucel Yemez. Slot-guided adaptation of pre-trained diffusion models for object-centric learning and compositional generation. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=kZvor5aaz7.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. doi: 10.1109/TPAMI.2013.50.

Usha Bhalla, Thomas Fel, Can Rager, Sheridan Feucht, Tal Haklay, Daniel Wurgaft, Siddharth Boppana, Matthew Kowal, Jack Merullo, Atticus Geiger, and Ekdeep Singh Lubana. Do sparse autoencoders capture concept manifolds? In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=euoAoOflaN.

Christopher P. Burgess, Loïc Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matthew M. Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *ArXiv*, abs/1901.11390, 2019.

Mathilde Caron, Hugo Touvron, Ishan Misra, Herv'e J'egou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9630–9640, 2021. URL https://api.semanticscholar.org/CorpusID:233444273.

Ricky T. Q. Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in vaes. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 2615–2625, Red Hook, NY, USA, 2018. Curran Associates Inc.

Stanislas Dehaene, Fosca Al Roumi, Yair Lakretz, Samuel Planton, and Mathias Sablé-Meyer. Symbols and mental programs: a hypothesis about human singularity. *Trends in Cognitive Sciences*, 26(9):751–766, 2022. ISSN 1364-6613. doi: https://doi.org/10.1016/j.tics.2022.06.010. URL https://www.sciencedirect.com/science/article/pii/S1364661322001413.

Aniket Didolkar, Andrii Zadaianchuk, Anirudh Goyal, Mike Mozer, Yoshua Bengio, Georg Martius, and Maximilian Seitzer. Zero-shot object-centric representation learning. *arXiv:2408.09162*, 2024. URL https://arxiv.org/abs/2408.09162.

Aniket Rajiv Didolkar, Andrii Zadaianchuk, Rabiul Awal, Maximilian Seitzer, Efstratios Gavves, and Aishwarya Agrawal. Ctrl-o: Language-controllable object-centric visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Nat Dilokthanakul, Pedro A. M. Mediano, Marta Garnelo, M. J. Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *ArXiv*, abs/1611.02648, 2016. URL https://api.semanticscholar.org/CorpusID:1327363.

Cian Eastwood and Christopher K. I. Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=By-7dz-AZ.

Gamaleldin Fathy Elsayed, Aravindh Mahendran, Sjoerd van Steenkiste, Klaus Greff, Michael Curtis Mozer, and Thomas Kipf. SAVi++: Towards end-to-end object-centric learning from real-world videos. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022.

Martin Engelcke, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. In *International Conference on Learning Representations*, 2020.

Martin Engelcke, Oiwi Parker Jones, and Ingmar Posner. GENESIS-v2: Inferring unordered object representations without iterative refinement. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021.

S. M. Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, koray kavukcuoglu, and Geoffrey E Hinton. Attend, infer, repeat: Fast scene understanding with generative models. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Thomas Fel, Matthew Kowal, Mozes Jacobs, Dron Hazra, Usha Bhalla, Lee Sharkey, Lucius Bushnaq, Satchel Grant, Tal Haklay, Thomas Icard, Can Rager, Michael Pearce, Daniel Wurgaft, Aiden Swann, Fenil Doshi, Siddharth Boppana, Curt Tigges, Nick Cammarata, Thomas Serre, Vasudev Shyam, Owen Lewis, Thomas McGrath, Jack Merullo, Ekdeep Singh Lubana, and Atticus Geiger. Structuring sparsity: Block-sparse featurizers capture visual concept manifolds, 2026a. URL https://arxiv.org/abs/2606.25234.

Thomas Fel, Binxu Wang, Michael A. Lepori, Matthew Kowal, Andrew Lee, Randall Balestriero, Sonia Joseph, Ekdeep Singh Lubana, Talia Konkle, Demba E. Ba, and Martin Wattenberg. Into the rabbit hull: From task-relevant concepts in DINO to minkowski geometry. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=pbbRmnFkjG.

Jerry A. Fodor and Zenon W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1):3–71, 1988. ISSN 0010-0277. doi: https://doi.org/10.1016/0010-0277(88)90031-5.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tcsZt9ZNKD.

Anirudh Goyal and Y. Bengio. Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 478, 10 2022. doi: 10.1098/rspa.2021.0068.

Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Klaus Greff, Raphael Lopez Kaufman, Rishabh Kabra, Nicholas Watters, Christopher P. Burgess, Daniel Zoran, Loïc Matthey, Matthew M. Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. *ArXiv*, abs/1903.00450, 2019.

Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks, 2020. URL https://arxiv.org/abs/2012.05208.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: a scalable dataset generator. 2022.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

Jindong Jiang*, Sepehr Janghorbani*, Gerard De Melo, and Sungjin Ahn. Scalor: Generative world models with scalable object representations. In *International Conference on Learning Representations*, 2020.

Jindong Jiang, Fei Deng, Gautam Singh, and Sungjin Ahn. Object-centric slot diffusion. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8563–8601, 2023.

Yibo Jiang, Goutham Rajendran, Pradeep Kumar Ravikumar, Bryon Aragam, and Victor Veitch. On the origins of linear representations in large language models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=otuTw4Mghk.

Zhu-Xi Jiang, Yin Zheng, Hua-Chun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. In *International Joint Conference on Artificial Intelligence*, 2016. URL https://api.semanticscholar.org/CorpusID:2546662.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1988–1997, 2016.

Whie Jung, Dong Hoon Lee, and Seunghoon Hong. Disentangled representation learning via modular compositional bias. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=x46uZ1HarC.

Laurynas Karazija, Iro Laina, and C. Rupprecht. Clevrtex: A texture-rich benchmark for unsupervised multi-object segmentation. *ArXiv*, abs/2111.10265, 2021. URL https://api.semanticscholar.org/CorpusID:244463087.

Ilyes Khemakhem, Diederik P. Kingma, and Aapo Hyvärinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, 2019. URL https://api.semanticscholar.org/CorpusID:195874364.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2649–2658. PMLR, 10–15 Jul 2018.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Thomas Kipf, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. In *International Conference on Learning Representations*, 2022.

Bohdan Kivva, Goutham Rajendran, Pradeep Ravikumar, and Bryon Aragam. Identifiability of deep generative models without auxiliary information. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Adam Kosiorek, Hyunjik Kim, Yee Whye Teh, and Ingmar Posner. Sequential attend, infer, repeat: Generative modelling of moving objects. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Jannik Kossen, Karl Stelzner, Marcel Hussing, Claas Voelcker, and Kristian Kersting. Structured object-aware physics prediction for video modeling and planning. In *International Conference on Learning Representations*, 2020.

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017. doi: 10.1017/s0140525x16001837.

Zhixuan Lin, Yi-Fu Wu, Skand Peri, Bofeng Fu, Jindong Jiang, and Sungjin Ahn. Improving generative imagination in object-centric world models. In *International Conference on Machine Learning*, 2020.

Hongjia Liu, Rongzhen Zhao, Haohan Chen, and Joni Pajarinen. Metaslot: Break through the fixed number of slots in object-centric learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=CJ00kttEjh.

Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Scholkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, 2018. URL https://api.semanticscholar.org/CorpusID:54089884.

Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020a.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 11525–11538. Curran Associates, Inc., 2020b.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff (eds.), *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 746–751, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://aclanthology.org/N13-1090/.

Bac Nguyen, Yuhta Takida, Naoki Murata, Chieh-Hsin Lai, Toshimitsu Uesaka, Stefano Ermon, and Yuki Mitsufuji. Improved object-centric diffusion learning with registers and contrastive alignment, 2026. URL https://arxiv.org/abs/2601.01224.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024a.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024b. URL https://proceedings.mlr.press/v235/park24c.html.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 76441–76463,

2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/be7430d22a4dae8516894e32f2fcc6db-Paper-Conference.pdf.

Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, and Francesco Locatello. Bridging the gap to real-world object-centric learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Rui Shu, Yining Chen, Abhishek Kumar, Stefano Ermon, and Ben Poole. Weakly supervised disentanglement with guarantees. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJgSwyBKvr.

Gautam Singh, Fei Deng, and Sungjin Ahn. Illiterate DALL-e learns to compose. In *International Conference on Learning Representations*, 2022a.

Gautam Singh, Yi-Fu Wu, and Sungjin Ahn. Simple unsupervised object-centric learning for complex and naturalistic videos. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 18181–18196. Curran Associates, Inc., 2022b.

Gautam Singh, Yeongbin Kim, Sungjin Ahn, and sdks. Neural systematic binder. In *The Eleventh International Conference on Learning Representations*, 2023.

Krishnakant Singh, Simone Scahub-Meyer, and Stefan Roth. Glass: Guided latent slot diffusion for object-centric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Wolfgang Stammer, Antonia Wüst, David Steinmann, and Kristian Kersting. Neural concept binder. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://openreview.net/forum?id=ypPzyflbYs.

Aleksandar Stanic and Jürgen Schmidhuber. R-sqair: Relational sequential attend, infer, repeat. *ArXiv*, abs/1910.05231, 2019. URL https://api.semanticscholar.org/CorpusID:204401838.

Karl Stelzner, Robert Peharz, and Kristian Kersting. Faster attend-infer-repeat with tractable probabilistic models. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 5966–5975, 2019. URL http://proceedings.mlr.press/v97/stelzner19a.html.

Matthew Trager, Pramuditha Perera, Luca Zancato, Alessandro Achille, Parminder Bhatia, and Stefano Soatto. Linear spaces of meanings: Compositional structures in vision-language models. pp. 15349–15358, 10 2023. doi: 10.1109/ICCV51070.2023.01412.

Zihao Wang, Lin Gui, Jeffrey Negrea, and Victor Veitch. Concept algebra for score-based conditional model. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023. URL https://openreview.net/forum?id=NHHonXNT26.

Nick Watters, Loic Matthey, Chris P. Burgess, and Alexander Lerchner. Spatial broadcast decoder: A simple architecture for disentangled representations in VAEs, 2019.

A.N. Whitehead. *Symbolism, Its Meaning and Effect*. Page-Barbour lectures. Macmillan, 1927.

Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. Slotformer: Unsupervised visual dynamics simulation with object-centric models. In *The Eleventh International Conference on Learning Representations*, 2023a.

Ziyi Wu, Jingyu Hu, Wuyue Lu, Igor Gilitschenski, and Animesh Garg. Slotdiffusion: Object-centric generative modeling with diffusion models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 50932–50958. Curran Associates, Inc., 2023b.

Tao Yang, Yuwang Wang, Yan Lu, and Nanning Zheng. Visual concepts tokenization. *ArXiv*, abs/2205.10093, 2022. URL https://api.semanticscholar.org/CorpusID:248965124.

Tao Yang, Yuwang Wang, Yan Lu, and Nanning Zheng. Disdiff: Unsupervised disentanglement of diffusion probabilistic models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=3ofe0lpwQP.

Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Object-centric learning for real-world videos by predicting temporal feature similarities. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

A Theoretical Proofs

A.1 Proof of identifiability theorem

Dropping the slot index for simplicity, and letting c denote the concatenated factor vector, the generative model for a slot can be written as:

$$s = f(\mathbf{c}) + \epsilon \quad \text{where} \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \Sigma_S) \tag{9}$$

Where $f(\mathbf{c}) = \sum_{i=1}^{M} f_i(c_i)$ is also a piecewise affine function. The distribution of the concatenated vector c is a joint GMM:

$$\mathbf{c} \sim \sum_{k=1}^{K} \bar{\pi}_k \mathcal{N}(\bar{\mu}_k, \bar{\Sigma}_k) \tag{10}$$

where the total number of components is $K = \prod_{i=1}^{m} K_i$. Here, $\bar{\mu}_k$ and $\bar{\Sigma}_k$ are formed by concatenating the corresponding component means $\mu_{k,i}$ and block-diagonalizing the covariances $\Sigma_{k,i}$ across all factors i, and $\bar{\pi}_k$ is the product of the relevant mixing weights $\pi_{i,k}$.

According to the theorem established by Kivva et al. (2022), the latent variable c in this linear Gaussian model is identifiable up to permutation, scaling, and translation. Suppose we have two trained models that yield the identical marginal distribution for s:

$$s \sim \mathcal{N}(f(\mathbf{c}), \Sigma) \qquad \text{where} \quad c \sim \sum_{k=1}^{K} \bar{\pi}_k \mathcal{N}(\bar{\mu}_k, \bar{\Sigma}_k) \tag{11}$$

$$s \sim \mathcal{N}(f'(\mathbf{c'}), \Sigma') \qquad \text{where} \quad c' \sim \sum_{k=1}^{K} \bar{\pi}'_k \mathcal{N}(\bar{\mu}'_k, \bar{\Sigma}'_k) \tag{12}$$

Identifiability implies that the latent variables from the two models are related by:

$$c = \Pi D c' + b \tag{13}$$

where Π is a permutation matrix, D is a diagonal scaling matrix, and b is a translation vector.

We now argue that Π must be a block-permutation matrix; that is, it cannot distribute the dimensions of a single factor vector c'^u from the second model across two different factors c^i and c^v in the first model.

Assume, for the sake of contradiction, that Π does split an factor. Without loss of generality, suppose a subset of dimensions from c'^u is assigned to c^i, and the remaining dimensions of c'^u are assigned to a different factor c^v.

Conversely, this means the factor c^i in the first model is composed of scaled, permuted, and translated dimensions sourced from at least two independent factors in the second model (e.g., c'^u and c'^w). Let these source factors have K_u and K_w mixture components, respectively.

Intuitively, combining dimensions from different independent factors necessitates a multiplicative expansion in the number of mixture components to properly model the new joint space (i.e., it requires $K_u \times K_w$ components). However, our target factor c^i is strictly defined as a sum over

only K_i components. Thus, mixing these dimensions would demand significantly more underlying components than the model's structural capacity allows.

Formally, because the source factors are independent, concatenating them forms a joint GMM with $K_u K_w$ components:

$$c^i \sim \sum_{l=1}^{K_u K_w} \pi_l'' \mathcal{N}(\mu_l'', \Sigma_l'') \tag{14}$$

However, by our initial definition, c^i itself is drawn from a GMM with exactly K_i components:

$$c^i \sim \sum_{l=1}^{K_i} \pi_{i,l} \mathcal{N}(\mu_{i,l}, \Sigma_{i,l}) \tag{15}$$

Equating these two distributions yields:

$$\sum_{l=1}^{K_i} \pi_{i,l} \mathcal{N}(\mu_{i,l}, \Sigma_{i,l}) \equiv \sum_{l=1}^{K_u K_w} \pi_l'' \mathcal{N}(\mu_l'', \Sigma_l'') \tag{16}$$

This equivalence requires $K_i = K_u K_w$. However, we explicitly assumed that $K_i \neq K_u K_w$ for any triplet of distinct indices (i, u, w). This is a contradiction. Therefore, Π cannot split factors across different blocks, proving that Π must be block-diagonal.

A.2 Derivation of the ELBO

Note that the generative process of the data is:

$$p_\theta(X, \mathbf{S}, \mathbf{C}) = p_{\theta_X}(X|\mathbf{S}) \prod_{i=1}^{N} \left(p_{\theta_S}(s_i|\mathbf{c}_i) \prod_{j=1}^{M} p_{\theta_{C_j}}(c_{j,i}) \right) \tag{17}$$

We further define following joint two distributions

$$p_{\theta_S}(\mathbf{S}|\mathbf{C}) = \prod_{i=1}^{N} p_{\theta_S}(s_i|\mathbf{c}_i) \quad \text{and} \quad p_{\theta_C}(\mathbf{C}) = \prod_{i=1}^{N} \left(\prod_{j=1}^{M} p_{\theta_{C_j}}(c_{j,i}) \right) \tag{18}$$

Further we also define the following joint distributions from variational posterior as:

$$q_{\phi_C}(\mathbf{C}|\mathbf{S}) = \prod_{i=1}^{N} q_{\phi_C}(\mathbf{c}_i|s_i) \tag{19}$$

The log likelihood of the data can be written as:

$$\log p_\theta(X) = \log \left(\int_{\mathbf{S}} \int_{\mathbf{C}} p_{\theta_X}(X|\mathbf{S}) p_{\theta_S}(\mathbf{S}|\mathbf{C}) p_{\theta_C}(\mathbf{C}) d\mathbf{S} d\mathbf{C} \right) \tag{20}$$

$$\leq \mathbb{E}_{q_\phi(\mathbf{S},\mathbf{C}|X)} \left[\log \left(\frac{p_{\theta_X}(X|\mathbf{S}) p_{\theta_S}(\mathbf{S}|\mathbf{C}) p_{\theta_C}(\mathbf{C})}{q_{\phi_S}(\mathbf{S}|X) q_{\phi_C}(\mathbf{C}|\mathbf{S})} \right) \right] \tag{21}$$

$$= \mathbb{E}_{q_{\phi_S}(\mathbf{S}|X)} \left[\log p_{\theta_X}(X|\mathbf{S}) \right] - \underbrace{D_{KL}\left(q_{\phi_S}(\mathbf{S}|X) \,\|\, p_{\theta_S}(\mathbf{S}|\mathbf{C}) \right)}_{\text{slot regularizer}} \tag{22}$$

$$- \underbrace{\mathbb{E}_{q_{\phi_S}(\mathbf{S}|X)} \left[D_{KL}\left(q_{\phi_C}(\mathbf{C}|\mathbf{S}) \,\|\, p_{\theta_C}(\mathbf{C}) \right) \right]}_{\text{block regularizer}}. \tag{23}$$

A.3 Concept Memory projection as Explicit block regularizer

The block regularizer encourages the inferred block distribution to match the GMM prior. This regularizer can be factored at the object level and further decomposed into a sum of regularizers

over individual blocks. We then use the result from Dilokthanakul et al. (2016); Jiang et al. (2016) for the KL divergence between a normal distribution and a GMM:

$$D_{KL}\left(q_{\phi_C}(\mathbf{C}|\mathbf{S}) \,\|\, p_{\theta_C}(\mathbf{C})\right) = \sum_{i=1}^{N} D_{KL}\left(q_{\phi_C}(\mathbf{c}_i|s_i) \,\|\, p_{\theta_C}(\mathbf{c}_i)\right) \tag{24}$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{M} D_{KL}\left(q_{\phi_C}(c_{j,i}|s_i) \,\|\, p_{\theta_{C_j}}(c_{j,i})\right) \tag{25}$$

Assuming that $q_{\phi_C}(c_{j,i}|s_i)$ is a normal distribution, each term in the above sum corresponds to the KL divergence between a normal distribution and a GMM. Using Jensen's inequality, this KL divergence can be upper-bounded by a weighted sum of KL divergences between the inferred distribution and the individual Gaussian components of the GMM.

$$D_{KL}\left(q_{\phi_C}(c_{j,i}|s_i) \,\|\, p_{\theta_{C_j}}(c_{j,i})\right) = -\mathbb{E}_{q_{\phi_C}(c_{j,i}|s_i)}\left[\log\left(\frac{p_{\theta_{C_j}}(c_{j,i})}{q_{\phi_C}(c_{j,i}|s_i)}\right)\right] \tag{26}$$

$$= -\mathbb{E}_{q_{\phi_C}(c_{j,i}|s_i)}\left[\log\left(\frac{\sum_{k=1}^{K_j}\pi_{k,j}\mathcal{N}(c_{j,i};\mu_{k,j},\Sigma_{k,j})}{q_{\phi_C}(c_{j,i}|s_i)}\right)\right] \tag{27}$$

$$\tag{28}$$

Introducing an auxiliary assignment distribution $\gamma_{k,j}(s_i)$ satisfying $\sum_k \gamma_{k,j}(s_i) = 1$, the above expression can be upper-bounded as:

$$D_{KL}\left(q_{\phi_C}(c_{j,i}|s_i) \,\|\, p_{\theta_{C_j}}(c_{j,i})\right) = -\mathbb{E}_{q_{\phi_C}(c_{j,i}|s_i)}\left[\log\left(\frac{\sum_{k=1}^{K_j}\pi_{k,j}\mathcal{N}(c_{j,i};\mu_{k,j},\Sigma_{k,j})\gamma_{k,j}(s_i)}{\gamma_{k,j}(s_i)q_{\phi_C}(c_{j,i}|s_i)}\right)\right] \tag{29}$$

$$\leq \gamma_{k,j}(s_i) D_{KL}(q_{\phi_C}(c_{j,i}|s_i)||\mathcal{N}(\mu_{k,j},\Sigma_{k,j})) \tag{30}$$

$$+ \sum_{k}^{K_j} D_{KL}(\gamma_{k,j}(s_i)||\pi_{k,j}) \tag{31}$$

Using the assumption that the $q_{\phi_C}(c_{j,i}|s_i)$ is normal distribution with mean $c_{j,i}$ and additionally that it have identity covariance matrix. Also assuming all the $\Sigma_{k,j}$ are also identity. Also ignoring the last term then the the best $\tilde{c}_{j,i}$ will be when the KL divergence is minimized.

$$c^*_{j,i} = \min_{c_{j,i}} \sum_{k=1}^{K_j} \gamma_{k,j}(s_i)||c_{j,i} - \mu_{k,j}||_2^2 \tag{32}$$

Which have the solution that

$$c^*_{j,i} = \sum_{k=1}^{K_j} \gamma_{k,j}(s_i)\mu_{k,j} \tag{33}$$

Thus, the optimal block representation is a weighted combination of the GMM prototype vectors. In particular, if the assignment weights are given by $\gamma_{k,j}(s_i) = \frac{\exp(\texttt{cosine-sim}(c_{j,i},\mu_{k,j}))}{\sum_k \exp(\texttt{cosine-sim}(c_{j,i},\mu_{k,j}))}$ and the prototype matrix is defined as $\mathcal{M}_j^T = \left[\mu_{1,j} \;..\; \mu_{K_j,j}\right]$ then the resulting representation corresponds to a projection of the block onto the prototype space.

This establishes the connection between the concept memory projection and the explicit block regularizer.

Table 4: Hyperparameters of training for CLEVR-Easy, CLEVR-Hard, and CLEVR-Tex datasets.

Module	Hyperparameter	Dataset		
		CLEVR Easy	CLEVR Hard	CLEVR Tex
General	Batch Size	64	64	64
	Image Size	128	128	128
	Training Steps	200K	260K	315K
	Encoder Learning Rate	0.0001	0.0001	0.0001
	Decoder Learning Rate (only used in phase 1)	0.0003	0.0003	0.0003
Block Refinement	Block Size (d_{block})	192	192	192
	# Blocks (N_b)	8	16	8
	# Concepts (K_j same across all j)	64	64	64
	# Refinement Steps (T_b)	3	3	3
	# Projections (N_p)	192	192	192
Slot Attention	Block Size (d_{slot})	192	192	192
	# Slots (N_s)	4	4	6
	# Refinement steps (T_s)	3	3	3

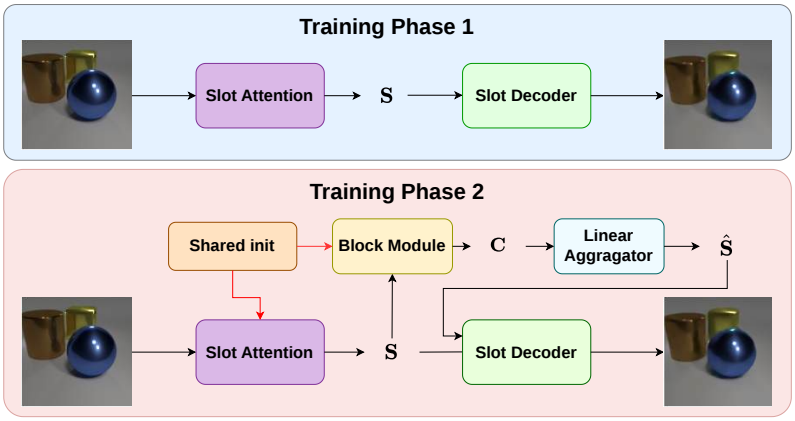


Figure 7: Training phases of the model.

B Experiment Details

B.1 Two Phase Training

Please refer to fig **??** where we show our two phase training pipeline. In phase one we train the slot attention and decoder on the slot method specific loss. In phase two we add the block specific modules and keep the decoder frozen.

B.2 Additional Details of the Datasets

We use three image benchmarks derived from Johnson et al. (2016); Karazija et al. (2021), provided by Singh et al. (2023). These datasets consist of the multi object images with 128×128 resolution. The objects in these datasets are characterized by set of categorical and continuous attributes with details as below. Additionally for slot linearity validation we further use MOVi-C Greff et al. (2022)

CLEVR-Easy dataset features scenes with 2-3 objects. Each object have 3 attributes: color with 8 possible choices, shape with 3 possible choices and position which is continuous.

CLEVR-Hard, a more challenging dataset with images consisting of 2-3 objects. Each object has 5 attributes: color with 137, shape with 3 and material with 2 possible choices; size comes from fixed range and the position is continuous.

Table 5: SLATE Results

METHOD	FG-ARI
SLATE	0.8166
SLATE+	0.9369

CLEVR-Tex dataset consists of textured images where the objects and background both have realistic textures increasing the visual complexity. The images consist of 2-3 objects with each object having 3 attributes. Texture/Material have 57 possible choices, shapes have 4 possible choices and position is continuous.

MOVi-C is the originally video dataset of objects rendered on the natural background. The frames from video as used as images.

B.3 CREATION OF THE PAIRED DATASET

Specifically we make a dataset, where each sample $t^{(l)}$ is tuple of four images $(x^{(l,1)}, x^{(l,2)}, x^{(l,3)}, x^{(l,4)})$ where both $(x^{(l,1)}, x^{(l,2)})$ share all the underlying objects except the one object. For simplicity assume it is the first object while generation in both the images. The objects differ in only one attribute say $i \in \{m\}$. That is $c_{1,j}^{(l,1)} = c_{1,j}^{(l,2)}$ for all $j \in \{1, ., m\} - \{i\}$. Similarly in $x^{(l,3)}$, $x^{(l,4)}$ all the objects are same except the first object. And those first objects only differ in i^{th} attribute, $c_{1,j}^{(l,3)} = c_{1,j}^{(l,4)}$ for all $j \in \{1, ., m\} - \{i\}$. Additionally the first object from $x^{(l,3)}$ shares the ith attribute with the first object from the $x^{(l,1)}$. That is $c_{1,i}^{1} = c_{1,i}^{3}$ and similarly for the $x^{(l,4)}$ and $x^{(l,2)}$, $c_{1,i}^{1} = c_{1,i}^{4}$. Let the corresponding slot representing these first objects be $s_j^{(l,1)}$ in the $x^{(l,1)}$ and $s_{j'}^{(l,2)}$ in the $x^{(l,2)}$. Here also we denote the corresponding slots representing the first objects from $x^{(l,3)}$ and $x^{(l,4)}$ as $s_k^{(l,3)}$ and $s_{k'}^{(l,3)}$.

C ADDITIONAL RESULTS

C.1 SLATE MODIFICATIONS

We modified the original SLATE Singh et al. (2022a) with initializing the slots as linear transformation of 2048 dimensional random vector sampled from learned gaussian rather than from just the learned normal distribution. Additionally we add a learnable linear projector on the slots before passing them to transformer slot decoder which makes 8 different projections of slots. The transformer decoder then uses this $8N$ tokens for decoding same as block decoder does. Refer table 5 which shows comparison on the CLEVR-Easy dataset.