An Object-Attribute Decoupled Approach for Learning Disentangled Representation for Image and Video Analysis

Sanket Gandhi [1] **Atul** [2] **Samamnyu Mahajan** [3] **Rushil Gupta** [4] **Vishal Sharma** [5 6] **Arnab Kumar Mondal** [3] **Rohan Paul** [3] **Parag Singla** [3]

Abstract

Learning disentangled representations for images and videos in terms of objects and their attributes without explicit supervision is an important but challenging task. Recent work (Singh et al., 2023) extends slot-based techniques for object discovery by decomposing slots into blocks, where each block is expressed as a linear combination of a fixed number of learnable concepts. At its core, this approach couples object and attribute discovery, assuming that image encoders innately learn disentangled features—an assumption we find does not always hold experimentally. We propose DeCoupler, a method that separates object discovery from attribute discovery by first using foundation models to extract object masks, and then learning block representations that capture attributes across objects. This leads to improved disentanglement, enabling tasks such as attribute-level interventions and dynamics prediction. We demonstrate these capabilities through experiments on five image and two video datasets, showing superior disentanglement and generalization over prior methods.

1. Introduction

In this paper, we ask the question whether modern-day AI systems can learn to decompose images/videos into building blocks made up of objects and their attributes, albeit without any explicit supervision? Recently, object-centric learning has shown promising results for decomposing scenes into objects via *slots* (Locatello et al., 2020). In this line of work, slots attend spatially over image features competing with each other, and are refined over iterations leading to discovery of objects. More recently, SysBinder (Singh et al., 2023) work with block-slot attention, where slots are now constrained to be composed of *blocks*, with blocks aimed at discovering meaningful attributes of objects. However, block-slot attention couples object discovery with attribute discovery, increasing the complexity of the learning task. Furthermore, each *block* is associated with an object corresponds to a specific region of the image encoding. This imposes a strong prior: that object attributes must always be recoverable directly from the image encoder's representation. Such a constraint assumes that the encoder alone can disentangle attribute information, which may not hold in general (Locatello et al., 2018; Hyvärinen & Pajunen, 1999). We propose an alternative method for learning a disentangled representation for objects and attributes, called DeCoupler. DeCoupler decouples the problem into one of *object discovery* and *attribute discovery*. We leverage pre-trained segmentation foundation models (Ravi et al., 2024) for object discovery, which require prompts to generate object masks. These prompts are obtained using slot-based methods and then passed to the foundation models for segmentation. In the second stage, the object masks thus generated are then passed to a *block-extractor* module, where blocks can be thought of representing object attributes. Blocks are obtained by having a *non-spatial* attention over object feature maps, which is refined over multiple iterations. In the style of (Singh et al., 2023), *blocks* are projected to a learnable concept space at the end of each refinement iteration. Since our attribute discovery module is not tied with the object discovery module, it can also be seen as plug-in-play type module and can be used on any object discovery method. We show that DeCoupler outperforms the existing baselines for discovering objects and their attributes of five image datasets and 2 video datasets. We also show the utility of DeCoupler in visual dynamic predictions where its block aware dynamics prediction approach outperforms existing object-centric approaches. *This opens doors to object- and attribute-level interventions learned in*

[*]Equal contribution [1]Yardi School of Artificial Intelligence, IIT Delhi, India [2]Department of Mathematics, IIT Delhi, India [3]Department of Computer Science and Engineering, IIT Delhi, India [4]Université de Montréal, Canada [5]Microsoft, Bengaluru, India [6]Work done while at IIT Delhi. Correspondence to: Sanket Gandhi <aiz238706@iitd.ac.in>, Atul <mt1210623@iitd.ac.in>, Rohan Paul <rohan@cse.iitd.ac.in>, Parag Singla <parags@cse.iitd.ac.in>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

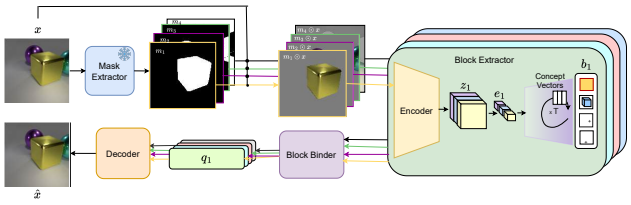

Figure 1. Given an input image x, we first extract object masks $m_i{}_{i=1}^{N}$ using a *mask extractor*. Each masked region (i.e., $x \odot m_i$) is then passed through a *block extractor* to obtain r blocks per object: $\{b_i^1, \ldots, b_i^r\}_{i=1}^N$. These blocks are subsequently processed by a *block binder*, which aggregates them into a single representation vector per object: $\{q_i\}_{i=1}^N$. Finally, these object-level vectors are fed into a decoder to reconstruct the image $\hat{x}$.

an unsupervised manner, directly aligning with the broader goals of the 'Scaling Up Intervention Models' workshop at ICML.

This paper makes the following contributions: (1) We propose DeCoupler, a novel decoupled approach for object and attribute discovery. (2) We propose novel block attention, a non-spatial attention algorithm to discover object attributes. (3) Our method can be applied to both image and video analysis, as well as video prediction as a downstream task. (4) We perform extensive evaluation over five image and two video datasets to the show the effectiveness of our approach.

2. DeCoupler

Our method DeCoupler consists of three components: (a) *Mask Extractor* which decomposes the image in the form of object*masks*. (b) *Block Extractor* which decoposes the object content to a set of *blocks*, where each block can be thought of as representing a object attribute. (c) *Decoder* which takes the block level representation of the image, and stitches it back to get the original image. We explain how this approach can be extended to videos by modeling the temporal component over the block representation. Figure 1 shows the overall pipeline of our approach. We next explain each part in detail.

Mask Extractor. To extract object masks from images, we first employ a slot-based model (Locatello et al., 2020), which discovers objects by attending over image feature maps and representing each object as a latent slot. The coarse slot-derived masks are then used to generate prompts for a pre-trained segmentation model (Ravi et al., 2024), which refines these into high-quality object masks. It is important to note that Slot Attention is trained beforehand, and the masks are extracted in a pre-processing step.

Given an image $x \in \mathbb{R}^{H \times W \times 3}$, we extract N slot-attention masks $\{u_1, .., u_N\}$ from a trained slot model where $u_i \in [0,1]^{H \times W}$ for $i \in \{1, .., N\}$. These slot attention masks are fed in a *Prompt generator* which for each slot-mask u_i, outputs the following two things (a) A set of positive points (locations) $p_i = \{p_{i_1}, .., p_{i_K}\}$, where each $p_{i_k} \in [H] \times [W]$ is obtained by choosing those points on masks which are surrounded by high mask value points. (b) A set of negative points n_i for each mask, where each $n_i = \bigcup_{i' \neq i} p_{i'}$ is simply the union of sampled positive points for other masks in the image. These two quantities with additional x_i as mask prompt are then provided to a pre-trained foundation model, such as SAM2, to obtain the N objects masks $\{m_i\}_{i=1}^N$ where each $m_i \in [0,1]^{H \times W}$. These N masks are subsequently fed into the *Block Attention* to obtain a factorized representation of the objects. Refer Appendix B for more details.

Algorithm 1 Block Attention: Inputs are object features $e_i \in \mathbb{R}^{L \times d_{in}}$ which are mapped to r blocks of dimension d_{block}. Model parameters are: linear projectors k, q and v with d_{block} output dimension; the block specific `GRU`s, and `MLP`s; initial blocks $b^1, .., b^r \in \mathbb{R}^{d_{block}}$; and concept vectors $C_1, ..., C_r \in \mathbb{R}^{k \times d_{block}}$

Initialize $b_i^j = b^j$ for all `j=1,...,r`
for `t = 1 to T` **do**
  b_i^j = `LayerNorm`(b_i^j) for all `j=1,..,r`
  A = `Softmax`$(\frac{1}{\sqrt{d_{block}}} q(b_i)k(e_i)^T$, `d='block')`
  $A = A$ / A`.sum(d='input')`
  $U = A.v(e_i)$
  for `j = 1 to r` **do**
    U_j = `GRU`$_j$`(state=`b_i^j`, input=`U_j`)`
    $U_j = U_j$ + `MLP`$_j(U_j)$
    w_i^j = `Softmax`$(\frac{1}{\eta\sqrt{d_{block}}} C_j U_j^T$`,d='concepts')`
    $b_i^j = C_j^T w_i^j$
  end for
end for
return b_i

Block Extractor. We disentangle the objects into what we call *blocks*. These *blocks* are obtained by iterative refinement with competitive attention over latent object representation which is jointly learned along with the block representation. The extracted masks m_i are multiplied element-wise with the corresponding input image x to get the content $c_i = m_i \odot x \in \mathbb{R}^{H \times W \times 3}$. This content is encoded through the CNN-based encoder f_ϕ to get the object latent representation as $z_i = f_\phi(c_i) \in \mathbb{R}^{H' \times W' \times L d_{in}}$. We take the mean of z_i along *spatial* axis and chunk it into L vector along *feature* axis denoted as row matrix $e_i \in \mathbb{R}^{L \times d_{in}}$. Given the latent object representation e_i for i^{th} mask, we denote the its block representation as $\{b_i^j\}_{j=1}^r, b_i^j \in \mathbb{R}^{d_{block}}$ where r is number of blocks and is a hyper-parameter of the model. Define $b_i = [b_i^1\ b_i^2\ ..\ b_i^r] \in \mathbb{R}^{r \times d_{block}}$. Algorithm 1 outlines the iterative refinement steps to obtain the block representation. Each b_i^j is initialized with the the learnable vector b^j. The block vectors then attend to object features to compute attention scores, which are made competitive through the *softmax* normalizing over queries (blocks). Attention scores are sum normalized over keys (object features). The resultant linear combination of object features is first passed

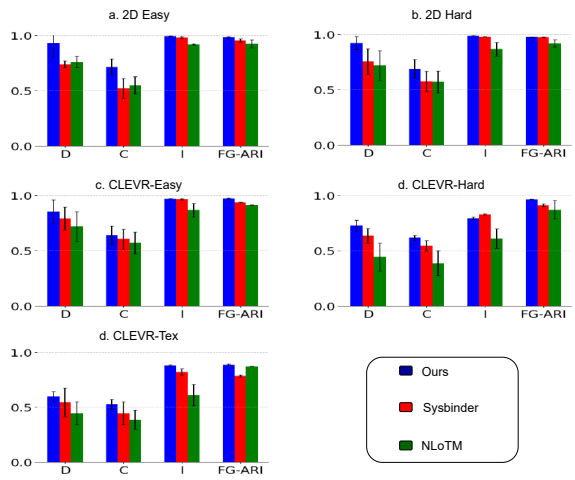

Figure 2. Results for various Approaches for Attribute Discovery

through a block-specific GRU, followed by an block-specific MLP to get the block representation. Finally, we project each resultant block vector onto the learnable concept space and update its representation as a linear combination of the concepts via projection weights (lines 10 - 11 in Algorithm 1). Refer to Appendix C for more details.

Decoding using Block Binder. For decoding the blocks into an image we first create a single vector representation of the blocks using *block binder*. This single vector can be treated as *slot* for decoding purposes. Following (Locatello et al., 2020) we use spatial mixture decoder to decode these single vector representation into an image. Specifically, r blocks $b_i = [b_i^1 \,..\, b_i^r]$ corresponding to i^{th} mask are fed to *block binder* $h_\theta : b_i \rightarrow q_i \in \mathbb{R}^{d_q}$ to get single vector q_i. Then spatial mixture decoder $g_\theta : q_i \rightarrow \pi_i', \mu_i$ where $\pi_i' \in \mathbb{R}^{H \times W}$ and $\mu_i \in \mathbb{R}^{H \times W \times 3}$. Then the π_i' are normalized to get the alpha mask as $\pi_i = \frac{\exp(\pi_i')}{\sum_{j=1}^{N} \exp(\pi_j')}$. The finally the reconstructed image is obtained as $\hat{x} = \sum_{i=1}^{N} \pi_i \odot \mu_i$. Refer Appendix D for more details.

As training objective, all of the method except the *mask extractor* is trained end-to-end with reconstruction objective.= with mean squared loss: $L = \frac{1}{HW} ||x - \hat{x}||_2^2$

3. Experiments

We investigate: (1) Can DeCoupler discover object attributes from images? (2) Can it swap attributes and googenerate novel concepts via recombination? (Results in Appendix G) (3) Can it extend to videos? (4) Are its object representations useful for modeling dynamics? (5) Which components are crucial for disentanglement? We briefly describe datasets, baselines, and metric.

Datasets: We evaluate DeCoupler on five image and two video datasets. The image datasets include *2D Easy* and *2D Hard*, simple 2D datasets inspired by (Watters et al., 2019), and *CLEVR-Easy*, *CLEVR-Hard*, and *CLEVR-Tex*, 3D datasets used in (Singh et al., 2023). The video datasets include *BS-Hard*, a video extension of *2D Hard*, and *OBJ3D* from (Lin et al., 2020a). Additional dataset details are provided in Appendix F.

Baselines: For image analysis, we compare DeCoupler with two baselines based on block-slot attention: Sysbinder (Singh et al., 2023) and NLoTM (Wu et al., 2024). For video analysis, we extend Sysbinder to video using one-step predictor (Baek et al., 2025). Additionally, we use SAVI (Wu et al., 2023a), a video extension of slot-attention as baselines. For video prediction, we use SlotFormer (Wu et al., 2023a) as baseline. Refer Suppl. I for training details.

Metrics: For image and video analysis, we use the DCI metric (Disentanglement, Completeness, and Informativeness) (Singh et al., 2023). We also report FG-ARI, which quantifies the object discovery. For the video prediction task, we use SSIM (Wang et al., 2004), PSNR, and LPIPS (Zhang et al., 2018) as our evaluation metrics.

3.1. Image analysis

Disentanglement Scores. Figure 2 shows the DCI scores for various approaches on the image datasets. Clearly, DeCoupler performs better than all baselines in all datasets on disentanglement and completeness scores, by a significant margin demonstrating its capability to discover the attributes effectively compared to the baselines. On Informativeness, DeCoupler is either better or competitive with SysBinder, except for CLEVR-Hard dataset, where it is marginally worse. We hypothesize this may be due to a weaker decoding model, compared to SysBinder. NLoTM performs worse on all metrics. FG-ARI score for DeCoupler is consistently better than all baselines for our approach. For detailed numbers, refer Appendix L.

Interventions via Block Swapping To visualize the disentangled object representations learned by DeCoupler, we provide the results for the swapping experiments. Specifically, we swap the blocks representing an attribute of two objects in an image and then decode these new object representations into a new image. To determine which attribute is represented by a block, we rely on its importance score (obtained while computing DCI) across the attributes. Figure 3 show the results of swapping across the datasets. We note that DeCoupler is able to discover attributes in a distinct set of blocks, and swap them correctly, for all the attributes.

3.2. Video Analysis

Disentanglement Scores. Table 1 shows the DCI results for all the baseline approaches on both video datasets. To

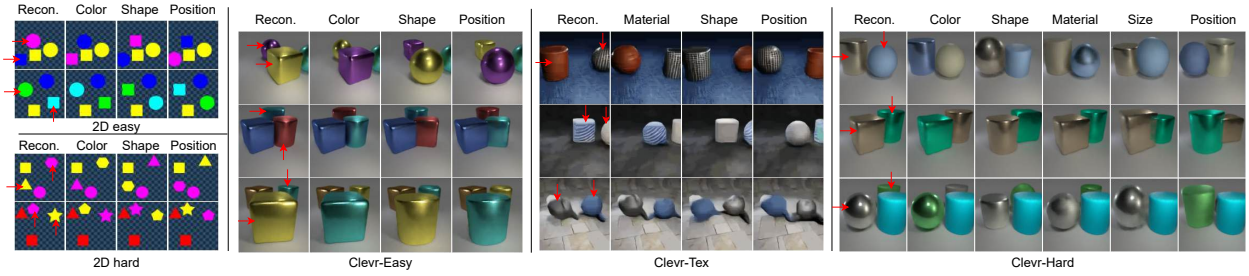

Figure 3. **Qualitative Results:** Swapping of attributes between objects. For each attribute, we pick the set of relevant blocks and swap them. The red arrows indicate the objects involved in swapping. *Recon.* shows the original image as reconstructed by DeCoupler.

compute DCI scores for videos, we first extract the blocks for the whole video. Next, we treat frames from all videos as images and adopt the same procedure for DCI computation as before. The training runs of Sysbinder for the OBJ3D video dataset on multiple random seeds, although converged, were not able to capture objects. Clearly, the simple video extension of DeCoupler outperforms all the baseline over all the datasets and metrics. We highlight that due to our decoupled approach DeCoupler is able to capture objects where as Sysbinder struggles to even discover the objects. Further, object attributes in the OBJ3D data set possess a slight bias. For example, the object that is launched is always a rubber sphere, and the stationary objects are always in the middle portion of the scene, making attribute discovery more challenging. However, despite such biases, the DeCoupler achieves high disentanglement performance as indicated by higher DCI scores in experiments. Refer Appendix H for swapping results.

Table 1. Comparison of DCI scores for videos

Model	BS			OBJ3D		
	D	C	I	D	C	I
SAVI	0.453	0.321	0.841	0.357	0.225	0.610
SysV	0.740	0.685	0.950	–	–	–
DecV	**0.988**	**0.773**	**0.998**	**0.740**	**0.685**	**0.951**

3.3. Visual Dynamics Learning

Prior work (Wu et al., 2023a;b) showed promising results for slot-based dynamics using transformers. We explore whether learning dynamics at the block level improves performance. We design two variants: *Blocks-SlotDyn*, which concatenates blocks as slots, and *Blocks-BlockDyn*, which uses blocks directly as tokens with a block-aware position encoder (Singh et al., 2023). As a control, we introduce *Denseformer*, which replaces blocks with MLP-encoded features. On the OBJ3D dataset Table 2, *Blocks-BlockDyn* outperforms all baselines, showing the benefit of block-level modeling. *Blocks-SlotDyn* also surpasses *Slotformer*, while *Denseformer* highlights the value of our object extractor.

Table 2. Comparison of video-prediction metrics

Model	LPIPS(↓)	SSIM(↑)	PSNR(↑)
Blocks-BlockDyn	**0.107**	**0.931**	**32.348**
Blocks-SlotDyn	0.112	0.929	32.159
Denseformer	0.147	0.919	31.223
Slotformer	0.204	0.919	30.801

3.4. Ablations

Table 3. Ablation on block extractor

Method	D	C	I
DeCoupler w/o attn	0.795	0.703	0.947
DeCoupler w/o con	0.862	0.605	0.937
DeCoupler w/o attn & con	0.203	0.175	0.740
DeCoupler	**0.890**	**0.662**	**0.949**

Block extractor has two components, attention and concept projection. We report the DCI scores obtained by dropping any one component and both. Table 3 shows the results obtained. It can be inferred that both the components are disentanglement enabler and removing any one of them drops the performance. Further removing feature attention of blocks drops the performance more than removing concept projection. This highlights the importance of feature attention with respect to concept projection for attribute discovery.

4. Conclusion and Future Work

In this paper, we have presented a decoupled approach for par disentanglement of both objects and attributes in images and videos. Our approach is based on a pipeline of an object-extractor utilizing pre-trained segmentation models, followed by a novel block-extractor which makes use of non-spatial attention over feature maps. Experiments demonstrate significant gains over existing baselines that are based on extensions of slot attention. This approach provides a mechanism learn disentangled object and attribute representations. This takes a step in the direction of object and attribute specific interventions relevant for image generation, model based planning etc.

References

Anonymous. Interaction asymmetry: A general principle for learning composable abstractions. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Anonymous. 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.

Anonymous. On the transfer of object-centric representation learning. In *The Thirteenth International Conference on Learning Representations*, 2025c.

Aydemir, G., Xie, W., and Guney, F. Self-supervised object-centric learning for videos. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Baek, J., Wu, Y.-F., Singh, G., and Ahn, S. Dreamweaver: Learning compositional world models from pixels. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=e5mTvjXG9u.

Biza, O., van Steenkiste, S., Sajjadi, M. S. M., Elsayed, G. F., Mahendran, A., and Kipf, T. Invariant slot attention: Object discovery with slot-centric reference frames. *ArXiv*, abs/2302.04973, 2023.

Brady, J., Zimmermann, R. S., Sharma, Y., Schölkopf, B., Von Kügelgen, J., and Brendel, W. Provably learning object-centric representations. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Burgess, C. P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M. M., and Lerchner, A. Monet: Unsupervised scene decomposition and representation. *ArXiv*, abs/1901.11390, 2019.

Chen, R. T. Q., Li, X., Grosse, R., and Duvenaud, D. 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.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 2180–2188, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Elsayed, G. F., Mahendran, A., van Steenkiste, S., Greff, K., Mozer, M. C., and Kipf, T. SAVi++: Towards end-to-end object-centric learning from real-world videos. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

Engelcke, M., Kosiorek, A. R., Jones, O. P., and Posner, I. Genesis: Generative scene inference and sampling with object-centric latent representations. In *International Conference on Learning Representations*, 2020.

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

Eslami, S. M. A., Heess, N., Weber, T., Tassa, Y., Szepesvari, D., kavukcuoglu, k., and Hinton, G. E. Attend, infer, repeat: Fast scene understanding with generative models. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Greff, K., Kaufman, R. L., Kabra, R., Watters, N., Burgess, C. P., Zoran, D., Matthey, L., Botvinick, M. M., and Lerchner, A. Multi-object representation learning with iterative variational inference. *ArXiv*, abs/1903.00450, 2019.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

Hyvärinen, A. and Pajunen, P. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439, 1999. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(98)00140-3. URL https://www.sciencedirect.com/science/article/pii/S0893608098001403.

Jiang*, J., Janghorbani*, S., Melo, G. D., and Ahn, S. Scalor: Generative world models with scalable object representations. In *International Conference on Learning Representations*, 2020.

Jiang, J., Deng, F., Singh, G., and Ahn, S. Object-centric slot diffusion. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8563–8601, 2023.

Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., and Girshick, R. B. 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.

Kakogeorgiou, I., Gidaris, S., Karantzalos, K., and Komodakis, N. Spot: Self-training with patch-order permutation for object-centric learning with autoregressive transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22776–22786, June 2024.

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

Kim, D., Kim, S., and Kwak, S. Bootstrapping top-down information for self-modulating slot attention. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Kim, H. and Mnih, A. Disentangling by factorising. In Dy, J. and Krause, A. (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.

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

Kipf, T., Elsayed, G. F., Mahendran, A., Stone, A., Sabour, S., Heigold, G., Jonschkowski, R., Dosovitskiy, A., and Greff, K. Conditional object-centric learning from video. In *International Conference on Learning Representations*, 2022.

Kori, A., Locatello, F., Santhirasekaram, A., Toni, F., Glocker, B., and Ribeiro, F. D. S. Identifiable object-centric representation learning via probabilistic slot attention. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

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

Kossen, J., Stelzner, K., Hussing, M., Voelcker, C., and Kersting, K. Structured object-aware physics prediction for video modeling and planning. In *International Conference on Learning Representations*, 2020.

Kumar, A., Sattigeri, P., and Balakrishnan, A. VARIATIONAL INFERENCE OF DISENTANGLED LATENT CONCEPTS FROM UNLABELED OBSERVATIONS. In *International Conference on Learning Representations*, 2018.

Lin, Z., Wu, Y.-F., Peri, S., Fu, B., Jiang, J., and Ahn, S. Improving generative imagination in object-centric world models. In *International Conference on Machine Learning*, 2020a.

Lin, Z., Wu, Y.-F., Peri, S. V., Sun, W., Singh, G., Deng, F., Jiang, J., and Ahn, S. Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. In *International Conference on Learning Representations*, 2020b.

Locatello, F., Bauer, S., Lucic, M., Gelly, S., Scholkopf, B., and Bachem, O. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, 2018. URL https://api.semanticscholar.org/CorpusID:54089884.

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. Object-centric learning with slot attention. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 11525–11538. Curran Associates, Inc., 2020.

Nakano, A., Suzuki, M., and Matsuo, Y. Interaction-based disentanglement of entities for object-centric world models. In *The Eleventh International Conference on Learning Representations*, 2023.

Nguyen, T., Mansouri, A., Madan, K., Khuong, N. D., Ahuja, K., Liu, D., and Bengio, Y. Reusable slotwise mechanisms. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=CniUitfEY3.

Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K. V., Carion, N., Wu, C.-Y., Girshick, R., Dollár, P., and Feichtenhofer, C. Sam 2: Segment anything in images and videos, 2024.

Seitzer, M., Horn, M., Zadaianchuk, A., Zietlow, D., Xiao, T., Simon-Gabriel, C.-J., He, T., Zhang, Z., Schölkopf, B., Brox, T., and Locatello, F. Bridging the gap to real-world object-centric learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Singh, G., Deng, F., and Ahn, S. Illiterate DALL-e learns to compose. In *International Conference on Learning Representations*, 2022a.

Singh, G., Wu, Y.-F., and Ahn, S. Simple unsupervised object-centric learning for complex and naturalistic videos. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in*

Neural Information Processing Systems, volume 35, pp. 18181–18196. Curran Associates, Inc., 2022b.

Singh, G., Kim, Y., and Ahn, S. Neural systematic binder. In *The Eleventh International Conference on Learning Representations*, 2023.

Singh, G., Wang, Y., Yang, J., Ivanovic, B., Ahn, S., Pavone, M., and Che, T. Parallelized spatiotemporal slot binding for videos. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Stammer, W., Wüst, A., Steinmann, D., and Kersting, K. Neural concept binder. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Tang, Q., Zhu, X., Lei, Z., and Zhang, Z. Intrinsic physical concepts discovery with object-centric predictive models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23252–23261, June 2023.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

Traub, M., Otte, S., Menge, T., Karlbauer, M., Thuemmel, J., and Butz, M. V. Learning what and where: Disentangling location and identity tracking without supervision. In *The Eleventh International Conference on Learning Representations*, 2023.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Veerapaneni, R., Co-Reyes, J. D., Chang, M., Janner, M., Finn, C., Wu, J., Tenenbaum, J., and Levine, S. Entity abstraction in visual model-based reinforcement learning. In Kaelbling, L. P., Kragic, D., and Sugiura, K. (eds.), *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1439–1456. PMLR, 30 Oct–01 Nov 2020.

Wang, Y., Liu, L., and Dauwels, J. Slot-vae: Object-centric scene generation with slot attention. *ArXiv*, abs/2306.06997, 2023.

Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13 (4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Watters, N., Matthey, L., Borgeaud, S., Kabra, R., and Lerchner, A. Spriteworld: A flexible, configurable reinforcement learning environment. https://github.com/deepmind/spriteworld/, 2019. URL https://github.com/deepmind/spriteworld/.

Wu, Y., Yoon, J., and Ahn, S. Generative video transformer: Can objects be the words? In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, Proceedings of Machine Learning Research, pp. 11307–11318. ML Research Press, 2021. Publisher Copyright: Copyright © 2021 by the author(s); 38th International Conference on Machine Learning, ICML 2021 ; Conference date: 18-07-2021 Through 24-07-2021.

Wu, Y.-F., Lee, M., and Ahn, S. Neural language of thought models. In *The Twelfth International Conference on Learning Representations*, 2024.

Wu, Z., Dvornik, N., Greff, K., Kipf, T., and Garg, A. Slotformer: Unsupervised visual dynamics simulation with object-centric models. In *The Eleventh International Conference on Learning Representations*, 2023a.

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

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

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. URL https://api.semanticscholar.org/CorpusID:4766599.

Zoran, D., Kabra, R., Lerchner, A., and Rezende, D. J. Parts: Unsupervised segmentation with slots, attention and independence maximization. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10419–10427, 2021.

A. Related Work

Unsupervised Object-Centric Learning aims to learn object representation from images or videos without any supervision. One line of work (Eslami et al., 2016; Lin et al., 2020b) represents objects from static images in latent z_{what} and z_{where} and uses spatial transformer decoder. This factorization is also extended to videos (Kosiorek et al., 2018; Jiang* et al., 2020; Lin et al., 2020a; Kossen et al., 2020; Traub et al., 2023). These methods are difficult to scale to complex datasets. Although these methods disentangle object representation into spatial position and content, unlike DeCoupler, they are not able to further disentangle the content into attributes. Another line of work is mixture-decoder-based methods (Burgess et al., 2019; Greff et al., 2019; Engelcke et al., 2020; 2021; Locatello et al., 2020; Biza et al., 2023; Wang et al., 2023) for images and (Kipf et al., 2022; Elsayed et al., 2022; Singh et al., 2024) for videos. Recently, it was observed that feature reconstruction objective enables to scale slot-based methods to the real world (Seitzer et al., 2023; Kakogeorgiou et al., 2024; Zadaianchuk et al., 2023; Aydemir et al., 2023; Kim et al., 2024; Anonymous, 2025c) for both images and videos. Diffusion as slot-decoder has also been explored for real world scaling of slots (Wu et al., 2023b; Jiang et al., 2023; Anonymous, 2025b). Recently there has been also work on assumptions needed to garuntee the object discovery (Brady et al., 2023; Kori et al., 2024; Anonymous, 2025a).

Disentangled Object Representation aims to to learn disentangled representation of objects. As mentioned above, one line of work just disentangles the objects into spatial z_{where} and entangled content z_{what}. The generative VAE and GAN based methods (Kingma & Welling, 2014; Higgins et al., 2017; Kim & Mnih, 2018; Chen et al., 2018; Kumar et al., 2018; Chen et al., 2016), are not able to disentangle images containing multiple objects. Recently block-slot attention have been propose to discover objects and attributes (Singh et al., 2023; Wu et al., 2024; Stammer et al., 2024; Baek et al., 2025)

Learning Visual Dynamics aims to learn underlying dynamics from videos. One line of work uses generative modeling of tasks (Kosiorek et al., 2018; Jiang* et al., 2020; Kossen et al., 2020; Lin et al., 2020a; Veerapaneni et al., 2020; Wu et al., 2021; Zoran et al., 2021). Slot-based methods which uses transformer for modeling dynamics have shown better results than generative modeling (Wu et al., 2023a;b; Nguyen et al., 2023; Tang et al., 2023). (Nakano et al., 2023) proposed method to learn the visual dynamics with z_{static} and $z_{dynamic}$ factorization of object.

B. Mask Extractor Additional Details

We observe that masks generated by slots, even though gives rough masks of object, can be improved. We use Segment Anything (SAM2)(Ravi et al., 2024) to further improve the mask quality. SAM is promptable image segmentation model, where prompt can be foreground / background points, a approximate mask or bounding box. We generate the points and mask prompt from masks generated by slots and pass it to SAM2 to get better mask.

We generate set of K foreground / positive point prompts per slot s_i denoted as p_i. A good choice for p_i is the points which are on the object and not on the boundary of the object. Our approach is to choose pixel positions with maximum pixel values. To make sure all points are well inside the object we first threshold the slot masks x_i with τ to make it binary mask m_i'. Then convolve m_i' with all one filter of size $(3, 3)$ for l times. Denote the resultant mask as $\tilde{M}_i$. We choose top K points which have maximum pixel value from $\tilde{M}_i$. For negative points we use the positive points of remaining slots which gives $(N-1)K$ negative / background points per slot. Along with this NK point prompt, we pass x_i as mask prompt to SAM. Algorithm 2 gives pseudo code for sampling point prompts from slot masks.

To handle the cases where slot does not represent object we ignore such slot masks and return all zero mask instead. If sum of pixel values of $\tilde{M}^i$ is less that `mthresh`, we consider that slot does not represent object. We use the official release of SAM2 with ViT-H backbone. [1].

[1] https://github.com/facebookresearch/sam2

Algorithm 2 Prompt Sampler: Inputs are: SAVi masks x_i where $i \in \{1, .., N\}$

for $i = 1$ to N **do**
  $M_i = x_i$.copy()
  $M_i[M_i \leq \tau] = 0$
  $M_i[M_i > \tau] = 1$
  for $j = 1$ to l **do**
    M_t^i =Convolve(input = M_i, kernel = all_1_3x3, padding = same)
  end for
  if $\tilde{M}_i$.sum() ¡ mthresh **then**
    p_i = None
  else
    p_i = argmax_topk($\tilde{M}_i$, k = K)
  end if
end for
return $p_1, .., p_{i_K}$

C. Block Extractor Additional Details

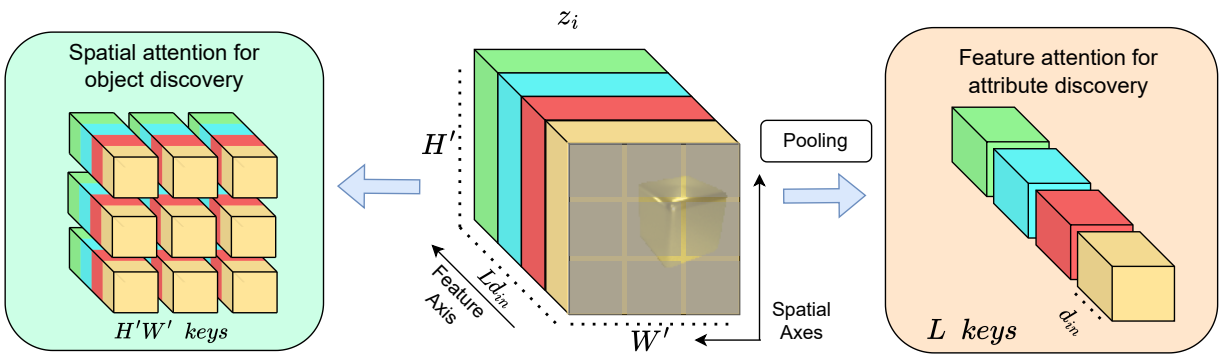

Figure 4. Left: Chunking of feature volume to create spatial features. Used by existing slot-based techniques and useful for object discovery. *Right*: Novel mechanism involving mean pooling of features and subsequent chunking, to obtain non-spatial global features. Useful for attribute discovery.

To develop some intuition for the framework in block extractor, we refer to Figure 4. The middle block in the figure shows 3-D representation of a latent object representation. We note that the latent representation has two axes of variation, one *spatial* axis and one *feature* axis figure 4. The key question is, if each block were to attend along either of the axis, which one would be more appropriate for attribute discovery? We argue here, that while for object discovery, it makes sense to attend along the spatial dimension, as is done by slot-attention based methods (Locatello et al., 2020), for attribute discovery, since we expect the attribute representation to be spread across the feature maps, it would make sense to attend along the feature axis. This is exactly what our key idea in this paper.

D. Block Binder Additional Details

The *block binder* is implemented as one layer transformer encoder (Vaswani et al., 2017), where the blocks are first linearly projected and then a block specific position encoding is added. The blocks, with a learnable token $q \in \mathbb{R}^{d_q}$ is fed into the transformer and final encoding of the q is treated as q_i. We also experimented with alternative transformer-based decoders (Singh et al., 2022a;b) but found that they performed similarly to the spatial mixture decoder except for visually complex datasets, albeit with added computational overhead. Diffusion based block decoders (Wu et al., 2023b; Jiang et al., 2023) could be explored in future work.

Note that above iterative refinement steps closely follows (Locatello et al., 2020) with the image latent replaced by object latent and slots replaced by blocks depicting factored (disentangled) object representation. Recently (Singh et al., 2023) proposed a similar-looking idea of learning disentangled representation for objects through block-slot attention. However, in block-slot attention, the blocks do not attend to object features, and the burden of disentanglement is placed on the encoder, making the learning process more challenging. In contrast, in our approach, blocks attend to object features during each forward pass, with no restrictions on the output of the encoder.

E. Video Extension

In this section we show that how our decoupled approach can be extended for disentangling video into objects and their attributes. For video the object representations needs to be temporally aligned, which means that in our decoupled approach *unsupervised mask extractor* needs to provide temporally aligned mask. This can be achieved by using an idea similar to slot models for video (Kipf et al., 2022; Elsayed et al., 2022; Wu et al., 2023a) as our object discovery model, and using the masks extracted from (Elsayed et al., 2022) as prompts to pre-trained segmentation models. For block extractor, since the static attributes of objects do not change within frames, we initialize the blocks at next time step with previous time step. Formally given sequence of frames $x_1, .., x_t$ the slot-based model will give temporally aligned slot-attention masks $f_s(\{x^l\}_{l=1}^t) = \{x_1^l, ..x_N^l\}_{l=1}^t$. Afterwards the *prompt generator* and segmentation foundation models treat each frame of sequence as image to get the final masks as $\{m_1^l, .., m_N^l\}_{l=1}^t$. Then in block attention each frame is treated same as image to get the blocks as $\{b_1^l, .., b_N^l\}_{l=1}^t$. The only change is in block initialization (line 1 of algorithm 1), where blocks are intialized with previous blocks as $b_i^{j,l} = pred_j(b_i^{j,l-1})$ where $pred_j$ is trainable block-specific function. The blocks at $l = 1$ are initialized with learnable vectors.

F. Datasets and Metrics

For image analysis, we use five image datasets described below. The first two datasets are inspired by (Watters et al., 2019) and created using (Todorov et al., 2012). Rest three datasets are inspired by (Johnson et al., 2016; Karazija et al., 2021) and are earlier used in (Singh et al., 2023; Wu et al., 2024; Stammer et al., 2024).

2D easy: Contains 4 objects (2 circles, 2 squares) with random colors (6 choices) and non-overlapping positions. The background has a checkerboard pattern.
2D hard: A more hard version of *2D-shape easy*. Objects (4 total) have random shapes and colors (6 choices each).
CLEVR-Easy: Images contain 2-3 objects with random shapes (3 choices), colors (8 choices), and positions.
CLEVR-Hard: A harder version of *CLEVR-Easy*, adding random size (fixed range) and material (2 choices). Colors (137 choices) and shapes (3 choices) are also randomized.
CLEVR-Tex: A harder *CLEVR-Easy* variant where objects have random shapes (4 choices), positions, and materials (57 choices).

For videos, we use two datasets:
Bouncing Shapes: A video extension of *2D easy* with elastic object collisions. Each 100-frame video contains 4 objects (2 circles, 2 squares) with colors chosen from 6 options.
OBJ3D: Used in (Lin et al., 2020a; Wu et al., 2023a), featuring a moving rubber sphere colliding with stationary objects. Static objects have randomized shape (3 choices), material (2 choices), size (3 choices), color (5 choices), and position.

Metrics: For image and video analysis, we use the DCI metric (Disentanglement, Completeness, and Informativeness) (Singh et al., 2023). We also report FG-ARI, which quantifies the object discovery. For the video prediction task, we use SSIM (Wang et al., 2004), PSNR, and LPIPS (Zhang et al., 2018) as our evaluation metrics.

G. Novel Concept Generation

Discovery of novel attributes is a potential application of learning disentangled representation for attributes. In this experiment, we evaluate whether novel values of attributes can be generated that have never been seen during training using DeCoupler .We first find the block representation of specific attribute values (e.g., *red color*) using ground truth data by taking the mean of object blocks representing that value of the attribute. Then, we make new color block representations by taking the weighted mean of two different color block representations of specific attribute values (e.g., *red & yellow color*) and decode the object representations with this new color block representation. Figure 5 shows the results of such generated images across the datasets and across the attributes. The decoded images show visual evidence of the novel object attributes appearing, *a-priori* unseen during training.

H. Swapping Results for Video

Similar to image analysis, we also perform swapping experiments on videos. Figure 6 shows the results on both the datasets. Specifically for video we swap the same latent block across frames. From the results, it is can be noted that DeCoupler is able to disentangle the attributes of the object in the Bouncing Shapes dataset. DeCoupler is able to maintain the same attribute

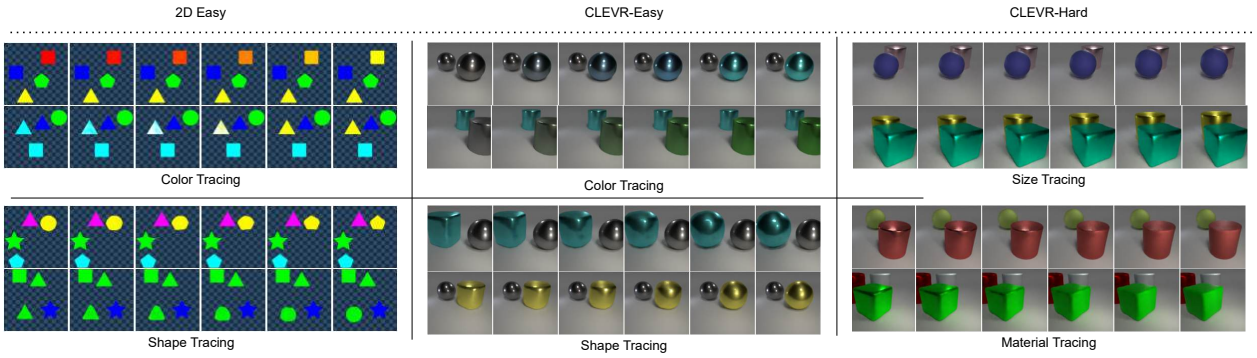

Figure 5. **Qualitative Results:** Generation of novel attribute instances. For each attribute, we have two instances, leftmost a_1 and rightmost a_2. The images in between show a linear combination, i.e. $a = \lambda a_1 + (1 - \lambda)a_2$. As we go from left to right, λ goes from 1 to 0. We observe that intermediate attributes are novel, and meaningful.

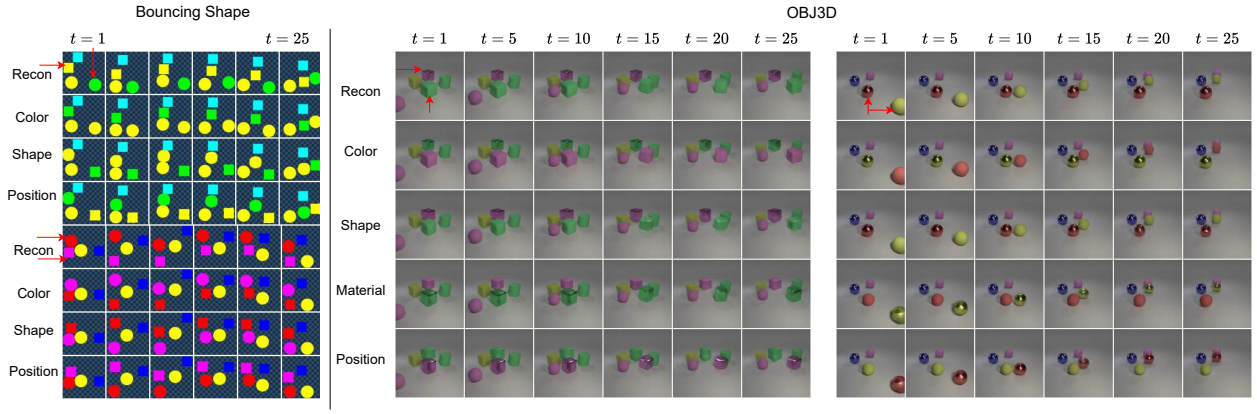

Figure 6. **Qualitative Results : Swapping** specific attributes between two objects.

in the same blocks across the time steps. Despite OBJ3D being challenging dataset due to biased attributes DeCoupler is able to swap the color, material as evident from Figure 6.

I. Implementation

I.1. Slot-Model for object extraction

For image datasets, we use two slot attention decoder variants based on dataset complexity. For visually simple datasets, we use the spatial broadcast decoder (Locatello et al., 2020), following the CLEVR6 setup. For more complex datasets, we adopt the transformer-based SLATE decoder (Singh et al., 2022a), using the configuration from (Singh et al., 2023). Table 4 lists the slot method and number of slots used for each dataset. For video datasets, we use the slot attention implementation from (Wu et al., 2023a), with hyperparameters kept the same as in the original OBJ3D setup. (Singh et al., 2023)[2]. For slate we keep the image size as 128 x 128. For SAVi we the implementation from (Wu et al., 2023a)[3]. We keep the image size as 64 x 64 and use the hyper parameters of OBJ3D from (Wu et al., 2023a).

I.2. Foundation Model to extract mask

And in case of video we just treat all frames form different videos as image dataset and use prompt sampler ans SAM2 in similar way as that for images.

We prompt SAM2 (Ravi et al., 2024) VIT-H to get refined object masks. The parameters used are given below.

[2]https://github.com/singhgautam/sysbinder
[3]https://github.com/pairlab/SlotFormer

Dataset	Slot-Model used	Number of Slots N
2D-Easy	Slot Attention	5
2D-Hard	Slot Attention	5
CLEVR-Easy	Slot Attention	4
CLEVR-Hard	Slot Attention	4
CLEVR-Tex	SLATE	4
Bouncing Shapes	SAVi	5
OBJ3D	SAVi	6

Table 4. Slot-models used for different datasets

	Dataset						
	2D-Easy	2D-Hard	CLEVR-Easy	CLEVR-Hard	CLEVR-Tex	Bouncing Shape	OBJ3D
K	1	1	1	1	1	1	1
τ	0.7	0.7	0.7	0.7	0.9	0.7	0.7
`mthresh`	45	45	45	45	45	45	45

Table 5. Prompt Sampler Hyperparameters for our model across different image datasets.

I.3. Hyperparameters For Image Analysis

I.3.1. 2D-EASY AND 2D-HARD

The encoder has same hyperparameter for both dataset.

1. Encoder f_ϕ

Layer	Kernel Size	Stride	Padding	Channels	Activation
Conv	5×5	2	2	32	ReLU
Conv	5×5	2	2	32	ReLU
Conv	5×5	2	2	32	ReLU
Conv	5×5	2	2	512	None

We use $L = 32$ and $d_{in} = 16$

2. Block Extractor:

3. Block Binder:

For 2D datasets we use the concatenation as attribute binder. Before concatenating we project the blocks to a smaller dimensional vector d_{down} with block specific projection matrix. $d_{down} = 4$ for 2D-Easy dataset and $d_{down} = 8$ for 2D-hard dataset. The output dimension for block binder is is $d_q = d_{down} \times r$.

4. Decoder:

I.3.2. CLEVR-EASY AND CLEVR-HARD

1. Encoder f_ϕ

We use $L = 128$ and $d_{in} = 32$

2. Block Extractor And Block Binder: Here $d_q = 128$.

3. Decoder:

Module	Hyperparameter	Dataset	
		2D-Easy	2D-Hard
General	Batch Size	64	64
	Training Steps	500K	500K
	Image Size	64 x 64	64 x 64
	Learning Rate	0.0002	0.0002
	Grad Clip	0.5	0.5
Block Extractor	Block Size (d_{block})	32	32
	# Blocks (r)	8	8
	# Prototypes (k)	32	32
	# Iterations (T)	3	3
	Temperature η	0.2	0.2

Table 6.

Type	Size/Channels	Activation	Comment
Spatial Broadcast	8×8	-	-
Position Embedding	-	-	
Conv 5×5	32	ReLU	stride: 2
Conv 5×5	32	ReLU	stride: 2
Conv 5×5	32	ReLU	stride: 2
Conv 5×5	4	None	stride: 1
Split Channels	RGB (3), alpha mask (1)	Softmax (on alpha masks)	-
Recombine	-	-	-

I.3.3. CLEVR-TEX

1. Encoder f_ϕ

We use $L = 128$ and $d_{in} = 32$

2. Block Extractor, Block Binder and Decoder: For CLEVR-tex we use transformer decoder which reconstruct the VQ-VAE tokens of image rather than pixels (Singh et al., 2022a). We use the hyper parameters of (Singh et al., 2023) to train VQVAE along with the DeCoupler . Here $d_q = 128$.

Layer	Kernel Size	Stride	Padding	Channels	Activation
Conv	5×5	2	2	64	ReLU
Conv	5×5	2	2	64	ReLU
Conv	5×5	2	2	64	ReLU
Conv	5×5	2	2	64	ReLU
Conv	5×5	2	2	4096	None

Module	Hyperparameter	Dataset	
		CLEVR-Easy	CLEVR-Hard
General	Batch Size	64	64
	Image Size	128 x 128	128 x 128
	Training Steps	500K	500K
	Learning Rate	0.0002	0.0002
	Grad Clip	0.5	0.5
Block Extractor	Block Size (d_{block})	256	128
	# Blocks (r)	8	16
	# Prototypes (k)	64	64
	# Iterations (T)	3	3
	Temperature η	0.2	0.2
Block Binder (Transformer Encoder)	# Layers	1	1
	# Heads	4	8
	Hidden Size	128	128
	Feed Forward Dim	256	256
	Dropout	0	0

Table 7.

I.4. Hyper Parameter for Video Analysis

The hyperparameter of encoder, decoder and attribute binder for bouncing shapes is same as that of 2D-easy dataset. The hyperparameter of encoder, decoder and attribute binder for OBJ3D are same as that of CLEVR-Hard. The block extractor is as

The one-step blocks predictor which is blocks specific is implemented as $pred_j(b_i^{j,l-1}) = b_i^{j,l-1} + \texttt{mlp}_j(b_i^{j,l-1})$. MLP is single hidden layer neural network with hidden dimension equal to that of block.

I.5. Hyper Parameter for Video prediction

Similar to (Wu et al., 2023a) we use only first 50 frames of video for training and testing. The dynamic models are trained to predict the future 10 frames give 6 past frames and tested to predict 44 future frames. We use the exact hyperparameters of (Wu et al., 2023a) with two difference: 1. we keep image size 128 rather than 64 and batch size 64 rather than 128.
The block aware dynamics module is just the transformer with additionally block coupler (Singh et al., 2023) to contextualize the blocks belonging to same object. The hidden dimension and number of layers of dynamics model are kept same as that

Type	Size/Channels	Activation	Comment
Spatial Broadcast	8×8	-	-
Position Embedding	-	-	
Conv 5×5	64	ReLU	stride: 2
Conv 5×5	64	ReLU	stride: 2
Conv 5×5	64	ReLU	stride: 2
Conv 5×5	64	ReLU	stride: 1
Conv 5×5	4	None	stride: 1
Split Channels	RGB (3), alpha mask (1)	Softmax (on alpha masks)	-
Recombine	-	-	-

Layer	Kernel Size	Stride	Padding	Channels	Activation
Conv	5×5	2	2	128	ReLU
Conv	5×5	2	2	128	ReLU
Conv	5×5	2	2	128	ReLU
Conv	5×5	2	2	128	ReLU
Conv	5×5	2	2	4096	None

Module	Hyperparameter	Dataset CLEVR-Tex
General	Batch Size	64
	Image Size	128 x 128
	Training Steps	500K
Block Extractor	Block Size (d_{block})	256
	# Blocks (r)	8
	# Prototypes (k)	64
	# Iterations (T)	3
	Temperature η	0.2
	Learning Rate	0.0002
	Grad Clip	0.5
Block Binder (Transformer Encoder)	# Layers	1
	# Heads	4
	Hidden Size	256
	Feed Forward Dim	512
	Dropout	0
VQVAE	Vocab Size	4096
	Learning Rate	0.0003
	Hidden Size	64
Decoder (Transformer Encoder)	# Layers	8
	# Heads	8
	Hidden Size	192
	Dropout	0.1

Table 8.

of (Wu et al., 2023a).

Module	Hyperparameter	Dataset	
		2D-Easy	OBJ3D
General	Batch Size	64	64
	Training Steps	500K	500K
	Image Size	64 x 64	128 x 128
	Learning Rate	0.0002	0.0002
	Grad Clip	0.5	0.5
	Sequence Length	6	6

Table 9.

J. Visual Results

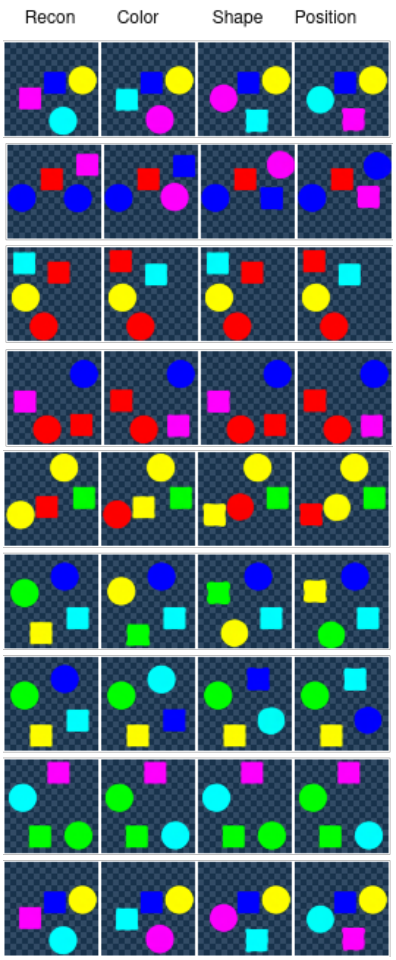

Figure 7. 2D easy swapping

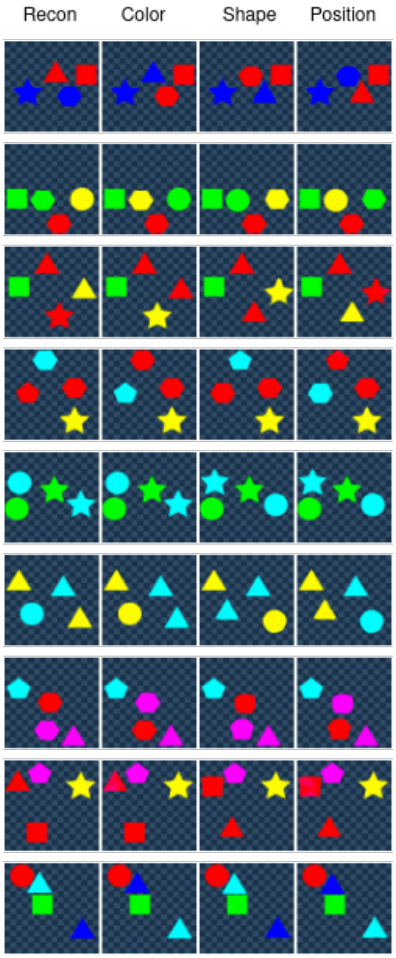

Figure 8. 2D hard swapping

K. Clusters

Here, we form the clusters on each block and visualize the samples assigned to a particular cluster. We see that on certain blocks, the clusters are formed on meaningful attributes of an object.

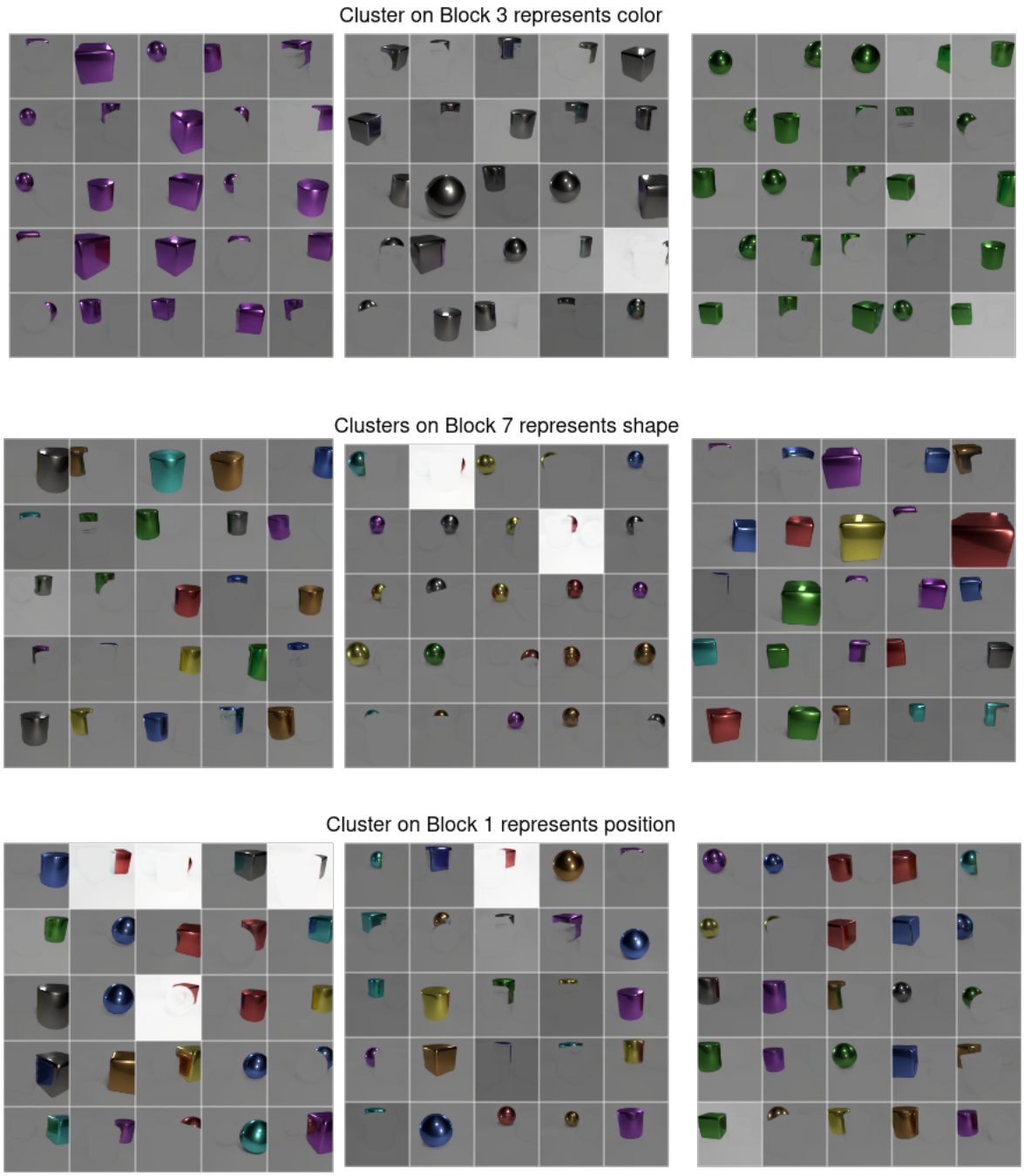

Figure 9. Block Clusters visualization

L. Additional Results

Model	D	C	I	FG-ARI
DeCoupler	$\mathbf{0.9338 \pm 0.1269}$	$\mathbf{0.7161 \pm 0.0736}$	$\mathbf{0.9936 \pm 0.0025}$	$\mathbf{0.9843 \pm 0.0080}$
Sysbinder	0.7391 ± 0.0308	0.5229 ± 0.0886	0.9815 ± 0.0089	0.9543 ± 0.0157
NLoTM	0.7612 ± 0.0510	0.5487 ± 0.0775	0.9206 ± 0.0058	0.9263 ± 0.0355

Table 10. Image metrics with standard deviation over 2D-Easy dataset

Model	D	C	I	FG-ARI
DeCoupler	$\mathbf{0.9234 \pm 0.0577}$	$\mathbf{0.6896 \pm 0.0852}$	$\mathbf{0.9890 \pm 0.0011}$	$\mathbf{0.9772 \pm 0.0007}$
Sysbinder	0.7569 ± 0.1156	0.5748 ± 0.0905	0.9790 ± 0.0020	0.9750 ± 0.0034
NLoTM	0.7201 ± 0.1349	0.5711 ± 0.0983	0.8679 ± 0.0603	0.9200 ± 0.0321

Table 11. Image metrics with standard deviation over 2D-Hard dataset

Model	D	C	I	FG-ARI
DeCoupler	$\mathbf{0.8547 \pm 0.1062}$	$\mathbf{0.6393 \pm 0.0844}$	$\mathbf{0.9700 \pm 0.0012}$	$\mathbf{0.9729 \pm 0.0050}$
Sysbinder	0.7945 ± 0.1040	0.6074 ± 0.0895	0.9654 ± 0.0050	0.9374 ± 0.0016
NLoTM	0.7201 ± 0.1349	0.5711 ± 0.0983	0.8679 ± 0.0603	0.9121 ± 0.0035

Table 12. Image metrics with standard deviation over CLEVR-Easy dataset

Model	D	C	I	FG-ARI
DeCoupler	$\mathbf{0.7268 \pm 0.0515}$	$\mathbf{0.6167 \pm 0.0244}$	0.7940 ± 0.0120	$\mathbf{0.9664 \pm 0.0005}$
Sysbinder	0.6361 ± 0.0668	0.5466 ± 0.0444	$\mathbf{0.8292 \pm 0.0075}$	0.9110 ± 0.0105
NLoTM	0.4451 ± 0.1251	0.3875 ± 0.1123	0.6116 ± 0.0886	0.8717 ± 0.0834

Table 13. Image metrics with standard deviation over CLEVR-Hard dataset

Model	D	C	I	FG-ARI
DeCoupler	$\mathbf{0.5970 \pm 0.0469}$	$\mathbf{0.5270 \pm 0.0466}$	$\mathbf{0.8809 \pm 0.0069}$	$\mathbf{0.8886 \pm 0.0079}$
Sysbinder	0.5460 ± 0.1316	0.4466 ± 0.1045	0.8234 ± 0.0287	0.7863 ± 0.0085
NLoTM	0.4451 ± 0.1049	0.3875 ± 0.0873	0.6116 ± 0.0979	0.8717 ± 0.0016

Table 14. Image metrics with standard deviation over CLEVR-Tex dataset