Learning Representations for Zero-Shot Image Generation without Text

Gautam Singh
Rutgers University
singh.gautam@rutgers.edu

Fei Deng
Rutgers University
fei.deng@rutgers.edu

Sungjin Ahn
Rutgers University
sungjin.ahn@rutgers.edu

Abstract

DALL·E has shown an impressive ability to generate novel—significantly and systematically different from the training distribution—yet realistic images. This is possible because it utilizes the dataset of text-image pairs where the text provides the source of compositionality. Following this result, an important extending question is whether this compositionality can still be achieved even without conditioning on text. In this paper, we propose a simple but novel slot-based autoencoding architecture, called SLATE[1], that achieves this text-free DALL·E by learning compositional slot-based representations purely from images, an ability lacking in DALL·E. Unlike existing object-centric representation models that decode pixels independently for each slot and each pixel location and compose them via mixture-based alpha composition, we propose to use the Image GPT decoder conditioned on the slots for a more flexible generation by capturing complex interaction among the pixels and the slots. In experiments, we show that this simple architecture achieves zero-shot generation of novel images without text and better quality in generation than the models based on mixture decoders.

1 Introduction

Unsupervised learning of compositional representation is a core ability of human intelligence (Yuille & Kersten, 2006; Frankland & Greene, 2020). Observing a visual scene, we perceive it not simply as a monolithic entity but as a geometric composition of key components such as objects, borders, and space (Kulkarni et al., 2015; Yuille & Kersten, 2006; Epstein et al., 2017; Behrens et al., 2018). Furthermore, this structured understanding about the scene composition enables the ability of *zero-shot imagination*, i.e., composing a novel, counterfactual, or systematically manipulated scenes which are significantly different from the training distribution, e.g., "what if I move the chair to the other room". As such, realizing this ability has been considered the core challenge in building a human-like AI system (Lake et al., 2017).

DALL·E (Ramesh et al., 2021) has recently shown an impressive result for zero-shot imagination. Trained with a dataset of text-image pairs, DALL·E can generate plausible images even from an unfamiliar text prompt such as "avocado chair" or "lettuce hedgehog". However, from the perspective of compositionality, the success of zero-shot imagination of DALL·E is arguably realizable with ease because of the fact that composable representation is already provided inherently in the form of the text prompt. That is, the text is already discretized into a sequence of composable concept modules, i.e., words, each of which is encapsulated as a reusable word vector. Given this, its Image GPT (Chen et al., 2020b) decoder learns to produce an image by smoothly stitching over the discretized concepts.

Extending from the success of DALL·E, an important question would probably be if we can achieve such zero-shot imagination only from images without text as we humans can do. This would require

[1]The implementation is available at `https://github.com/singhgautam/slate`

1st CtrlGen: Controllable Generative Modeling in Language and Vision Workshop at NeurIPS 2021.

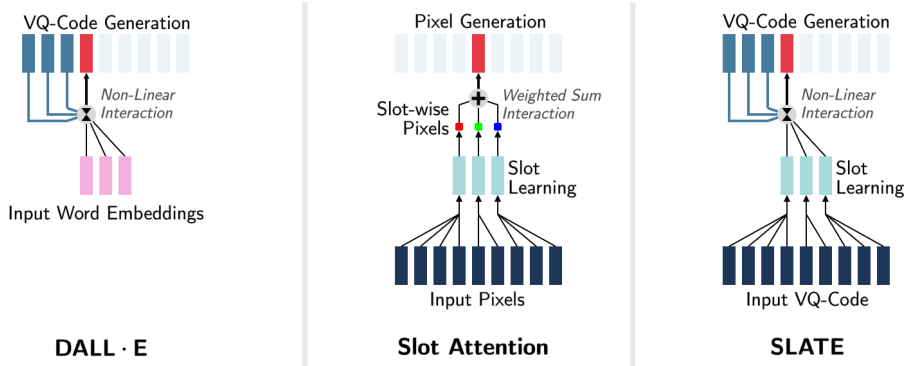


Figure 1: Overview of our proposed model with respect to prior works. Left: In DALL·E, words in the input text act as the composable units for generating the desired novel image. The generated images have global consistency because each pixel depends non-linearly on all previous pixels and the input word embeddings. **Middle:** Unlike DALL·E that requires text supervision, Slot Attention provides an auto-encoding framework in which object slots act as the composable units inferred purely from raw images. However during rendering, object slots are composed via a simple weighted sum of pixels obtained without any dependency on other pixels and slots which harms image consistency and quality. **Right:** Our model combines the best of both models. Like Slot Attention our model is free of text-based supervision and like DALL·E, produces novel image compositions with global consistency.

an ability lacking in DALL·E: extracting a set of composable representation from an image. The most relevant approach toward this direction is object-centric representation learning (Greff et al., 2019; Locatello et al., 2020; Lin et al., 2020b; Jiang et al., 2019; Chen et al., 2020a). While it can obtain a set of slots from an input image and reconstruct the same image from the slots, its ability to *reconfigure* an arbitrary set of slots for zero-shot imagination is significantly limited. We find that this is due to the mixture-based decoder used in these models which decodes each slot and pixel rather independently.

In this paper, we propose a simple architecture that can combine the best of DALL·E and object-centric representation learning. That is, like DALL·E, our model can do zero-shot imagination but by learning a set of slot representations from the input image without relying on text prompt. The key idea is simple: to combine the slot-attention with the Image GPT decoder used in DALL·E. We call the proposed model SLot Attention TransformEr or SLATE. In the experiments, we show that this simple architecture achieves the zero-shot generation of novel images but without text, better quality in generation than the models based on mixture decoders, and also provides structured relational representation

The contribution of the paper is a new architecture that can be seen as either a text-free DALL·E or an object-centric representation and generation model that significantly improves the systematic out-of-distribution generalization ability in image generation. Our result suggests for the first time that an Image GPT decoder can be used for object-centric representation learning, making it much simpler and more effective than previous approaches.

2 Background

2.1 Object-Centric Learning and Current Limitations in Compositional Generation

Object-centric learning is commonly done via an auto-encoding framework (Locatello et al., 2020; Burgess et al., 2019) in which an encoder takes an input image and returns a set of object representations or slots $\mathbf{s}_{1:N} = f_\phi(\mathbf{x})$. The slots are then provided to a decoder that composes the objects represented by the slots and reconstructs the image $\hat{\mathbf{x}} = g_\theta(\mathbf{s}_{1:N})$. In these models, the architecture of the decoder g_θ implements some form of domain knowledge about how the object slots are composed to produce the final output image. The most common composition approach assumes that the generated image is a pixel-wise weighted mean of image components (Locatello et al., 2020; Burgess et al., 2019; Lin et al., 2020b). To do this, the decoder first decodes the image component

$\boldsymbol{\mu}_n$ and the corresponding masks π_n for each slot n as follows:

$$\boldsymbol{\mu}_n = g_\theta^{\text{RGB}}(\mathbf{s}_n) \qquad\qquad \boldsymbol{\pi}_n = \frac{\exp g_\theta^{\text{mask}}(\mathbf{s}_n)}{\sum_{m=1}^{N} \exp g_\theta^{\text{mask}}(\mathbf{s}_m)}$$

where g_θ^{RGB} and g_θ^{mask} decode the RGB component and the mask weights, respectively, for a specific object. These are then combined using a pixel-wise weighted mean to produce the final image as $\hat{\mathbf{x}}_i = \sum_{n=1}^{N} \pi_{n,i} \cdot \boldsymbol{\mu}_{n,i}$ where $i \in [1, HW]$ is the pixel index and H and W are the image height and width, respectively. While such alpha composition may work for simple synthetic images, this model suffers in generation quality when we attempt to do object composition in more realistic images. The key failure mode occurs because each object component $g_\theta^{\text{RGB}}(\mathbf{s}_n)$ and its mask weight $g_\theta^{\text{mask}}(\mathbf{s}_m)$ are decoded for each slot independently without incorporating the information about other slots. Because of this, when object slots selected from different input images are composed, the rendered image may become incoherent. For instance, when objects and their shadows are modeled as separate components, novel combinations of slots may result in shadows that become inconsistent with the object.

In this work, we shall lift the use of inductive biases such as alpha composition for rendering the scenes by making use of Transformer (Vaswani et al., 2017) as our image decoder. Our central hypothesis is that a powerful auto-regressive decoder such as Transformer after training with sufficiently diverse set of raw images should learn the rules of composition implicitly without domain specifications.

2.2 Image GPT and DALL·E

Image GPT (Chen et al., 2020a) is a generative model for images implemented using Transformer (Vaswani et al., 2017; Brown et al., 2020). To train the model, Image GPT first down-scales the image of size $H \times W$ by a factor of K using a VQ-VAE encoder (van den Oord et al., 2017). This makes the training of Transformer less costly. Thus an image $\mathbf{x}$ becomes a sequence of image tokens $\{\mathbf{z}_i\}$ where i indexes the tokens in a raster-scan order from 1 to HW/K^2. The transformer is trained to model an auto-regressive distribution over this token sequence by maximizing the log-likelihood $\sum_i \log p_\theta(\mathbf{z}_i|\mathbf{z}_{<i})$. During generation, the transformer samples the image tokens sequentially. This process may be described as $\hat{\mathbf{z}}_i \sim p_\theta(\hat{\mathbf{z}}_i|\hat{\mathbf{z}}_{<i})$ where $\hat{\mathbf{z}}_i$ is the image token generated for the position i in the sequence. Once all the tokens for the sequence are generated, these tokens $\{\hat{\mathbf{z}}_i\}$ are provided to a VQ-VAE decoder to generate the image $\hat{\mathbf{x}}$.

It has been shown that Image GPT performs accurate image generation and completion given the upper half of the image. Because it is able to generate consistent images containing multiple objects, this suggests that Image GPT has implicitly learned "object models" about how to render those individual objects and how to compose multiple objects together such that the generated image is consistent with the true underlying image generating process.

DALL·E (Ramesh et al., 2021) shows that this image generating ability of Image GPT can be controlled using text prompts. DALL·E models the distribution over the tokens of the image conditioned on the tokens of the text prompt. The process for generating the image given the text may be described as $\hat{\mathbf{z}}_i \sim p_\theta(\hat{\mathbf{z}}_i|\hat{\mathbf{z}}_{<i}, \mathbf{c}_{1:N})$ where $\mathbf{c}_{1:N}$ are the representations of N tokens of the text prompt on which the image generation is conditioned and $\hat{\mathbf{z}}_i$ is the image token at position i. However, in this approach for controlling image generation, a key limitation is that it requires supervision in terms of text-image pairs. In our work, instead of using text supervision, we shall show that composable object representations can be inferred directly from raw images without any supervision and these representations can be used to condition the image generation of an Image GPT decoder. By composing arbitrary object representations obtained from arbitrary input images, we show that our decoder has the ability to render novel scenes such that the generated image is consistent with the true underlying image generating process.

3 SLATE: Slot Attention Transformer

Our aim is to infer object-centric representations or simply slots from an input image and use the slots to condition the Transformer decoder to reconstruct the image as a sequence of pixels.

Obtaining Image Tokens using DVAE. To make the training of transformer computationally feasible for high-resolution images, we first downscale the input image $\mathbf{x}$ of size $H \times W$ by a factor of K

using Discrete VAE (Im et al., 2017). To do this, we split the image into $K \times K$-sized patches resulting in T patches where $T = HW/K^2$. We provide each patch $\mathbf{x}_i$ as input to an encoder network f_ϕ to return log probabilities (denoted as $\mathbf{o}_i$) for a categorical distribution with V classes. With these log probabilities, we use a relaxed categorical distribution (Jang et al., 2016) with a temperature τ to sample a relaxed one-hot vector $\mathbf{z}_i^{\text{soft}}$. We then decode these codes to obtain reconstructions of the patches.

$$\begin{aligned}
\mathbf{o}_i &= f_\phi(\mathbf{x}_i), \\
\mathbf{z}_i^{\text{soft}} &\sim \text{RelaxedCategorical}(\mathbf{o}_i; \tau), \\
\tilde{\mathbf{x}}_i &= g_\theta(\mathbf{z}_i^{\text{soft}}).
\end{aligned}$$

By minimizing an MSE reconstruction objective for the image patches i.e. $\mathcal{L}_{\text{DVAE}} = \sum_{i=1}^{T}(\tilde{\mathbf{x}}_i - \mathbf{x}_i)^2$, we train the DVAE encoder $f_\phi(\cdot)$ and decoder $g_\theta(\cdot)$ networks.

Inferring Object Slots. To infer the object slots from a given image $\mathbf{x}$, we first use the DVAE encoder as described above to obtain an image token $\mathbf{z}_i$ for each patch i. Next, we map each code $\mathbf{z}_i$ to an embedding by using a learned dictionary. To incorporate the position information into these patch embeddings, we add learned positional embeddings. This results in an embedding $\mathbf{u}_i$ which now has both the content and the position information. These embeddings $\mathbf{u}_{1:T}$ are then given as input to a Slot-Attention encoder (Locatello et al., 2020) with N slots.

$$\begin{aligned}
\mathbf{o}_i &= f_\phi(\mathbf{x}_i), \\
\mathbf{z}_i &\sim \text{Categorical}(\mathbf{o}_i), \\
\mathbf{u}_i &= \text{Dictionary}_\phi(\mathbf{z}_i) + \mathbf{p}_{\phi,i}, \\
\mathbf{s}_{1:N}, A_{1:N} &= \text{SlotAttention}_\phi(\mathbf{u}_{1:T}).
\end{aligned}$$

This results in N object slots $\mathbf{s}_{1:N}$ and N attention maps $A_{1:N}$ from the last refinement iteration of the Slot-Attention encoder.

Reconstruction using Transformer. To reconstruct the input image, we use the slots $\mathbf{s}_{1:N}$ to first reconstruct the DVAE code $\hat{\mathbf{z}}_{1:T}$ using a Transformer. We then use the DVAE decoder g_θ to decode the DVAE code and reconstruct the image patches $\hat{\mathbf{x}}_i$.

$$\begin{aligned}
\hat{\mathbf{o}}_i &= \text{Transformer}_\theta(\hat{\mathbf{u}}_{<i}; \mathbf{s}_{1:N}), \\
\hat{\mathbf{z}}_i &= \arg\max_{v \in [1,V]} \hat{o}_{i,v}, \\
\hat{\mathbf{x}}_i &= g_\theta(\hat{\mathbf{z}}_i).
\end{aligned}$$

where $\hat{\mathbf{u}}_i = \text{Dictionary}_\phi(\hat{\mathbf{z}}_i) + \mathbf{p}_{\phi,i}$ and $\hat{o}_{i,v}$ is the log probability of token v among the V classes of the categorical distribution represented by $\hat{\mathbf{o}}_i$. As the training objective for transformer, we minimize the cross-entropy of predicting each token $\mathbf{z}_i$ given all the preceding tokens $\mathbf{z}_{<i}$ and the slots $\mathbf{s}_{1:N}$. Let the predicted log-probabilities for the token at position i be $\bar{\mathbf{o}}_i = \text{Transformer}_\theta(\mathbf{u}_{<i}; \mathbf{s}_{1:N})$. Then the cross-entropy objective can be written as $\mathcal{L}_{\text{ST}} = \sum_{i=1}^{T} \text{CrossEntropy}(\mathbf{z}_i, \bar{\mathbf{o}}_i)$.

Learning Objective and Training. The complete training objective is given by $\mathcal{L} = \mathcal{L}_{\text{ST}} + \mathcal{L}_{\text{DVAE}}$ and all modules of our model are trained jointly. We apply a decay on the DVAE temperature τ from 1.0 to 0.1 at the start of the training. We also apply a learning rate warm-up for the parameters of Slot-Attention and Transformer at the start of the training as we found that this led to stable training.

3.1 Concept Library

The above model extracts a set of concepts or slots that describe an image whereas an intelligent agent would build a *library of reusable concepts* from diverse experience. In DALL·E, a vocabulary of words with learned embeddings plays the role of this library and provides reusable concepts even though the concepts or the words are pre-defined rather than constructed from experience.

To build a library of concepts from experience, in our experiments, we shall use the following simple approach based on K-means clustering to construct a library of reusable visual concepts. (i) Collect N slots $\mathcal{S}_i = \{\mathbf{s}_1^i, \ldots, \mathbf{s}_N^i\}$ and their attention maps $\mathcal{A}_i = \{A_1^i, \ldots, A_N^i\}$ from each image i in the training dataset. (ii) Run K-means clustering on $\mathcal{S} = \bigcup_i \mathcal{S}_i$ with cosine similarity between slots as a distance metric where K is the number of total concepts in the library. In cases when object position

is more important for building the concept library, it might be more natural to cluster the slots via IOU between the attention maps as a distance metric. (iii) Use each cluster as a concept and the slots assigned to the cluster as the instantiations of the concept. (iv) To compose an arbitrary image, choose the concepts from the library and randomly choose a slot for each concept. (v) Provide this set of slots to the decoder to compose and generate an image. Here, choosing concept vectors is like composing a text prompt in DALL·E. But in our case, our prompt is an order-less set of slots.

Considering that an agent may collect new experience (or images) indefinitely, it is an interesting future direction to scale this idea to large K and apply online clustering (Liberty et al., 2016) or Bayesian non-parametric clustering (Neal, 2000).

3.2 Multi-headed Slot Attention

For representing images, slots of standard Slot Attention encoder (Locatello et al., 2020) can suffer in expressiveness when representing objects with complex shapes and textures. This is because the slots collect information from the input cells via dot-product attention in which the attended input values are pooled using a simple weighted mean. As such, this pooling method can be too weak for representing complex objects. To address this, we propose an extension of Slot Attention called *Multi-headed Slot Attention* in which each slot attends to the input cells via multiple heads. This allows each slot to attend to different parts of the same object. When the slot state is computed, the attended values of different heads are concatenated which allows these partial object features to interact more flexibly and produce a significantly better object representation. Full details of the implementation and an ablation experiment to show the benefits of multiple heads are provided in Appendix A.2.

4 Related Work

Compositional Generation via Object-Centric Learning. Self-supervised object-centric approaches (Burgess et al., 2019; Greff et al., 2019; Locatello et al., 2020; Kabra et al., 2021; Greff et al., 2017; Engelcke et al., 2020, 2021; Eslami et al., 2016; Crawford & Pineau, 2019b; Lin et al., 2020b; Jiang & Ahn, 2020; Kosiorek et al., 2018; Jiang et al., 2019; Crawford & Pineau, 2019a; Lin et al., 2020a; Deng et al., 2021; Anciukevicius et al., 2020; von Kügelgen et al., 2020; Wu et al., 2021) typically perform mixture-based alpha compositing for rendering with contents of each slot decoded independently. Savarese et al. (2021) and Yang et al. (2020) rely on minimizing mutual information between predicted object segments as a learning signal while Yang et al. (2021) leverage optical flow. However, these approaches either cannot compose novel scenes or require specialized losses for object discovery unlike ours which uses a simple reconstruction loss. DINO (Caron et al., 2021) combining Vision Transformer (Dosovitskiy et al., 2021) and a self-supervised representation learning objective (Chen et al., 2020b; He et al., 2020; Grill et al., 2020; Touvron et al., 2021; Hinton, 2021; van den Oord et al., 2018) discovers object-centric attention maps but, unlike ours, cannot compose novel scenes. TIMs (Lamb et al., 2021) attach an image encoder (Goyal et al., 2021) to an Image GPT decoder and show object-centric attention maps but, unlike ours, do not investigate compositional generation.

Compositional Generation using GANs. Chai et al. (2021) show compositional generation using a collage of patches encoded and decoded using a pretrained GAN. However, the collage needs to be provided manually and object discovery requires a specialized process unlike our model. Bielski & Favaro (2019) and Chen et al. (2019) show object discovery from given images by randomly redrawing or adding a random jitter (Voynov et al., 2021) to the foreground followed by an adversarial loss. As these models use alpha compositing, their compositional ability is limited. Some works (Donahue et al., 2016; Donahue & Simonyan, 2019) infer abstract representations for images that emphasize the high-level semantics, but these, unlike ours, cannot be used for compositional image generation. Niemeyer & Geiger (2021), Nguyen-Phuoc et al. (2020); Chen et al. (2016), van Steenkiste et al. (2020), Liao et al. (2020) and Ehrhardt et al. (2020) introduce GANs which generate images conditioned on object-wise or factor-wise noise and optionally on camera pose. Lacking an inference module, these cannot perform compositional editing of a given image unlike ours. While Kwak & Zhang (2016) provide an encoder to discover objects, the decoding relies on alpha compositing. Reed et al. (2016a) show controlled compositional generation in GANs and improved generation quality (Johnson et al., 2018; Hinz et al., 2019) via an object-centric or a key-point based

Dataset	FID (↓)	
	SA	Ours
Shapestacks	155.74	**51.27**
Bitmoji	71.43	**15.83**

(a) Compositional Generation

Dataset	MSE (↓)		FID (↓)	
	SA	Ours	SA	Ours
Shapestacks	233.72	**111.86**	139.72	**30.22**
Bitmoji	388.72	**261.10**	67.66	**11.89**

(b) Image Reconstruction

Table 1: Comparison of Compositional Generation (left) and Image Reconstruction (right) between Slot-Attention (SA) and our model. For comparison of compositional generation, we report FID score. For image reconstruction quality, we report MSE and FID score.

pathway for scene rendering. However, these require supervision for the bounding boxes and the keypoints.

Compositional Generation with Latent Variable Models and Text. An early line of approaches focused on disentangling the independent factors of the observation generating process using β-VAE (Higgins et al., 2017) or ensuring that independently sampling each latent factor should produce images indistinguishable from the training distribution (Kim & Mnih, 2018; Kumar et al., 2017; Chen et al., 2018). However unlike ours, these approaches can suffer in multi-object scenes (Jiang & Ahn, 2020) due to *superposition catastrophe* (Greff et al., 2020). Another line of approaches learn to map text to images to compose novel scenes (Ramesh et al., 2021; Reed et al., 2016b; Li et al., 2019; Higgins et al., 2018) or map attribute-value pairs to images (Sohn et al., 2015; Yan et al., 2016). However such approaches rely on supervision.

5 Experiments

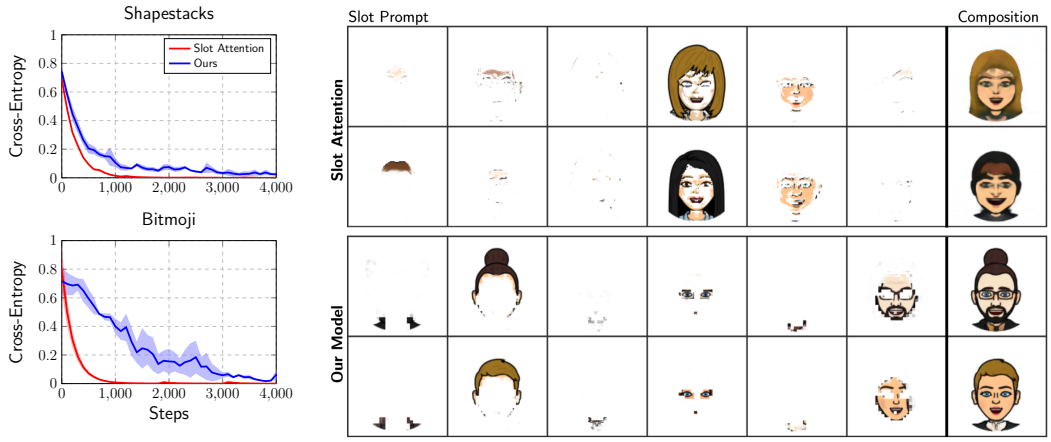
In experiments, we evaluate whether a Transformer decoder instead of a mixture decoder can provide benefits in: 1) generating novel scenes from unseen combinations of slots, 2) image reconstruction, and 3) generating novel scenes that do not belong to the training image distribution. For this, we compare our model with Slot Attention (Locatello et al., 2020) which, like ours, uses a slot-based encoder but uses a mixture decoder for decoding. We evaluate the models on 2 datasets: Shapestacks (Groth et al., 2018) and Bitmoji (Graux, 2021). In each dataset, a scene can be described in terms of composable entities such as blocks, walls, floors, hair or face. Our models take only raw images as inputs without any other supervision or annotations.

5.1 Compositional Image Generation.

In DALL·E, unseen text prompts were used to generate novel scenes. Similarly, in our text-free model, we evaluate whether novel scene generation is possible by providing unseen combinations of slots as the prompt for our Image GPT decoder.

To build slot prompts, we first build a concept library using K-means as described in Section 3.1. In Bitmoji, we apply K-means clustering on the slots. This results in clusters for *dress*, *hair*, *neck*, *eyes*, *face* and *collar*. These clusters make up our concept library and are visualized in Appendix C. To build novel prompts from this library, we simply pick one slot from each cluster. In Shapestacks dataset, object positions are more important and thus we slightly modify the above approach to build the prompts. We first sample a set of random positions on the canvas in a vertical tower configuration. Then for each of these positions, we then sample a slot from the concept library. In this way, we generate 40000 prompts for each dataset and render the corresponding images. We then evaluate the realness of the images by computing FID score with respect to the true images. To support this metric, we also report the training curves of a CNN discriminator that tries to classify between real and model-generated images. If generated images are close to the true data distribution, then they should be harder for the discriminator to discriminate and the resulting training curve should converge more slowly.

Better Visual Quality. From Figure 2a and Table 1a, we note that our compositional generations are significantly more realistic than the generations from the mixture decoder. This is also supported by qualitative samples shown in Figures 2b and 4.



(a) Discriminator Training Curves **(b)** Compositional Generation Samples in Bitmoji

Figure 2: Comparison of compositional generation between Slot Attention and our model. Left: Comparison of training curves of a discriminator to compare the quality of compositional generation between Slot-Attention and our model. A CNN discriminator receives either a real image or a model-generated image and tries to classify between real and fake. A slow converging curve is better. **Right:** We visualize the set of slots that the model receives as prompt and we visualize the composed scene returned by the model. We visualize the slots in the prompt by showing source image masked using the input attention map of that slot. Note that these attention maps had emerged in the source images implicitly without any supervision. We also note the limitations of mixture decoder of Slot Attention which produces incoherent compositions. For instance, the white background from the face slot can incorrectly mask the hair slot which is not the desired composition.

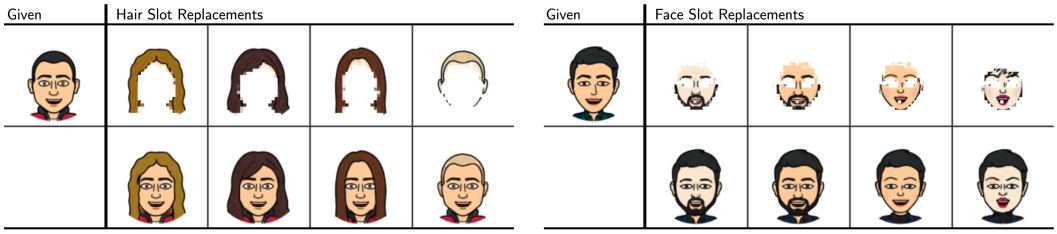


Figure 3: Scene Editing by Replacing Slots in Bitmoji. We show that in our model, it is possible to edit the image by taking a specific slot and replacing it with an arbitrary slot drawn from the concept library for the same concept.

Better Global Consistency. In Figure 2b, we visualize the compositions for Bitmoji datasets. These results show that the mixture decoder is unable to consistently combine the slots provided in the prompt. In Bitmoji, the mixture components of the source images interact incorrectly leading to inaccurate compositions. In comparison, our model is significantly more robust to this inconsistency problem when performing novel compositions.

Compositional Scene Editing. In our model, it is also possible to edit a given scene by inferring the slot representation and then replacing the slot with another one taken from the concept library. In Figure 3 (and Figure 7 in Appendix), we show compositional editing on Bitmoji dataset.

5.2 Reconstruction Quality

A natural test about whether an auto-encoder learns accurate representations of the given input is through an evaluation of the reconstruction error. For this, we report two metrics: 1) MSE to evaluate how well the reconstructed image preserves the contents of the original image and 2) FID score to show how realistic is the reconstructed image with respect to the true image distribution. We compute these metrics on a held-out set and report in Table 1b.

We find that our model outperforms Slot Attention in all datasets in terms of FID and MSE which shows that the image quality of our model is better. We also note qualitatively that Slot Attention

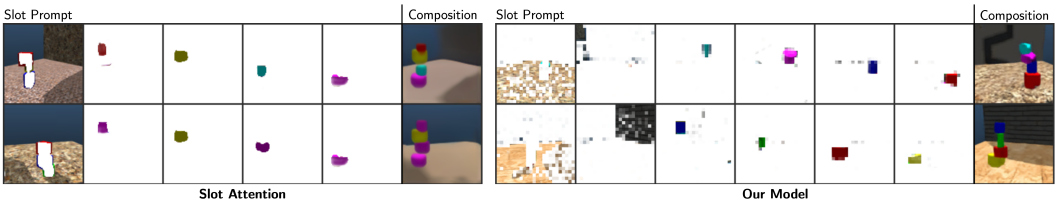


Figure 4: Compositional generation in Shapestacks Dataset. We provide 4 object slots and the background slots in the slot prompt and generate compositions.

renders more blurry images with fewer details than ours. One reason for this is Spatial Broadcast decoder (Watters et al., 2019) which is a weak decoder used for rendering object components in Slot Attention. Prior works use Spatial Broadcast decoder to bias the decoder towards simple object components and encourage disentanglement of objects into separate slots. However, our results show that such inductive biases are not needed and we do not need to trade-off image quality with object discovery (see Appendix D.1).

6 Conclusion

We presented a model for zero-shot imagination by learning slot representations from the input image. Our model combines the best of DALL·E and object-centric representation learning. It achieves novel scene composition without text while also providing significant improvement in generation quality over mixture-based decoders. Because we do not make any design choices specific to images, it is an interesting direction to explore our model on other domains such as text or audio.

7 Broader Impact

The current version of the model does not generate images realistic enough for negative societal impact. However future versions of the model with more computation, larger datasets, and better tuning may generate images with such impact. However, this is not imminent.

References

Anciukevicius, T., Lampert, C. H., and Henderson, P. Object-centric image generation with factored depths, locations, and appearances. *arXiv preprint arXiv:2004.00642*, 2020.

Behrens, T. E., Muller, T. H., Whittington, J. C., Mark, S., Baram, A. B., Stachenfeld, K. L., and Kurth-Nelson, Z. What is a cognitive map? organizing knowledge for flexible behavior. *Neuron*, 100(2):490–509, 2018.

Bielski, A. and Favaro, P. Emergence of object segmentation in perturbed generative models. In *Advances in Neural Information Processing Systems*, volume 32, pp. 7254–7264, 2019.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

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

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *ICCV 2021 - International Conference on Computer Vision*, 2021.

Chai, L., Wulff, J., and Isola, P. Using latent space regression to analyze and leverage compositionality in gans. In *ICLR 2021: The Ninth International Conference on Learning Representations*, 2021.

Chen, C., Deng, F., and Ahn, S. Object-centric representation and rendering of 3d scenes. *arXiv preprint arXiv:2006.06130*, 2020a.

Chen, M., Artières, T., and Denoyer, L. Unsupervised object segmentation by redrawing. In *Advances in Neural Information Processing Systems 32 (NIPS 2019)*, volume 32, pp. 12705–12716, 2019.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I. Generative pretraining from pixels. In *International Conference on Machine Learning*, pp. 1691–1703. PMLR, 2020b.

Chen, M., Radford, A., Child, R., Wu, J. K., Jun, H., Luan, D., and Sutskever, I. Generative pretraining from pixels. In *ICML 2020: 37th International Conference on Machine Learning*, volume 1, pp. 1691–1703, 2020a.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *ICML 2020: 37th International Conference on Machine Learning*, volume 1, pp. 1597–1607, 2020b.

Chen, T. Q., Li, X., Grosse, R. B., and Duvenaud, D. K. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pp. 2610–2620, 2018.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: interpretable representation learning by information maximizing generative adversarial nets. In *NIPS'16 Proceedings of the 30th International Conference on Neural Information Processing Systems*, volume 29, pp. 2180–2188, 2016.

Crawford, E. and Pineau, J. Exploiting spatial invariance for scalable unsupervised object tracking. *arXiv preprint arXiv:1911.09033*, 2019a.

Crawford, E. and Pineau, J. Spatially invariant unsupervised object detection with convolutional neural networks. In *Proceedings of AAAI*, 2019b.

Deng, F., Zhi, Z., Lee, D., and Ahn, S. Generative scene graph networks. In *ICLR 2021: The Ninth International Conference on Learning Representations*, 2021.

Donahue, J. and Simonyan, K. Large scale adversarial representation learning. In *Advances in Neural Information Processing Systems*, volume 32, pp. 10541–10551, 2019.

Donahue, J., Krähenbühl, P., and Darrell, T. Adversarial feature learning. In *ICLR (Poster)*, 2016.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR 2021: The Ninth International Conference on Learning Representations*, 2021.

Ehrhardt, S., Groth, O., Monszpart, A., Engelcke, M., Posner, I., Mitra, N., and Vedaldi, A. Relate: Physically plausible multi-object scene synthesis using structured latent spaces, 2020.

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

Engelcke, M., Jones, O. P., and Posner, I. Genesis-v2: Inferring unordered object representations without iterative refinement. *arXiv preprint arXiv:2104.09958*, 2021.

Epstein, R. A., Patai, E. Z., Julian, J. B., and Spiers, H. J. The cognitive map in humans: spatial navigation and beyond. *Nature neuroscience*, 20(11):1504–1513, 2017.

Eslami, S. A., Heess, N., Weber, T., Tassa, Y., Szepesvari, D., and Hinton, G. E. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems*, pp. 3225–3233, 2016.

Frankland, S. M. and Greene, J. D. Concepts and compositionality: in search of the brain's language of thought. *Annual review of psychology*, 71:273–303, 2020.

Goyal, A., Lamb, A., Hoffmann, J., Sodhani, S., Levine, S., Bengio, Y., and Schölkopf, B. Recurrent independent mechanisms. In *ICLR 2021: The Ninth International Conference on Learning Representations*, 2021.

Graux, R. Bitmoji dataset. https://www.kaggle.com/romaingraux/bitmojis/metadata/, 2021.

Greff, K., van Steenkiste, S., and Schmidhuber, J. Neural expectation maximization. In *Advances in Neural Information Processing Systems*, pp. 6691–6701, 2017.

Greff, K., Kaufmann, R. L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M., and Lerchner, A. Multi-object representation learning with iterative variational inference. *arXiv preprint arXiv:1903.00450*, 2019.

Greff, K., van Steenkiste, S., and Schmidhuber, J. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21271–21284, 2020.

Groth, O., Fuchs, F. B., Posner, I., and Vedaldi, A. Shapestacks: Learning vision-based physical intuition for generalised object stacking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 724–739, 2018.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9729–9738, 2020.

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 *ICLR 2017 : International Conference on Learning Representations 2017*, 2017.

Higgins, I., Sonnerat, N., Matthey, L., Pal, A., Burgess, C. P., Bosnjak, M., Shanahan, M., Botvinick, M., Hassabis, D., and Lerchner, A. Scan: Learning hierarchical compositional visual concepts. In *International Conference on Learning Representations*, 2018.

Hinton, G. E. How to represent part-whole hierarchies in a neural network. *arXiv preprint arXiv:2102.12627*, 2021.

Hinz, T., Heinrich, S., and Wermter, S. Generating multiple objects at spatially distinct locations. In *International Conference on Learning Representations*, 2019.

Im, D. I. J., Ahn, S., Memisevic, R., and Bengio, Y. Denoising criterion for variational auto-encoding framework. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *ICLR (Poster)*, 2016.

Jiang, J. and Ahn, S. Generative neurosymbolic machines. In *Advances in Neural Information Processing Systems*, 2020.

Jiang, J., Janghorbani, S., De Melo, G., and Ahn, S. Scalor: Generative world models with scalable object representations. In *International Conference on Learning Representations*, 2019.

Johnson, J., Gupta, A., and Fei-Fei, L. Image generation from scene graphs. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1219–1228, 2018.

Kabra, R., Zoran, D., Erdogan, G., Matthey, L., Creswell, A., Botvinick, M., Lerchner, A., and Burgess, C. P. Simone: View-invariant, temporally-abstracted object representations via unsupervised video decomposition. *arXiv preprint arXiv:2106.03849*, 2021.

Kim, H. and Mnih, A. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

Kosiorek, A., Kim, H., Teh, Y. W., and Posner, I. Sequential attend, infer, repeat: Generative modelling of moving objects. In *Advances in Neural Information Processing Systems*, pp. 8606–8616, 2018.

Kulkarni, T. D., Whitney, W., Kohli, P., and Tenenbaum, J. B. Deep convolutional inverse graphics network. *arXiv preprint arXiv:1503.03167*, 2015.

Kumar, A., Sattigeri, P., and Balakrishnan, A. Variational inference of disentangled latent concepts from unlabeled observations. In *International Conference on Learning Representations*, 2017.

Kwak, H. and Zhang, B.-T. Generating images part by part with composite generative adversarial networks. *arXiv preprint arXiv:1607.05387*, 2016.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

Lamb, A., He, D., Goyal, A., Ke, G., Liao, C.-F., Ravanelli, M., and Bengio, Y. Transformers with competitive ensembles of independent mechanisms. In *arXiv e-prints*, 2021.

Li, W., Zhang, P., Zhang, L., Huang, Q., He, X., Lyu, S., and Gao, J. Object-driven text-to-image synthesis via adversarial training. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12174–12182, 2019.

Liao, Y., Schwarz, K., Mescheder, L., and Geiger, A. Towards unsupervised learning of generative models for 3d controllable image synthesis. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5871–5880, 2020.

Liberty, E., Sriharsha, R., and Sviridenko, M. An algorithm for online k-means clustering. In *2016 Proceedings of the eighteenth workshop on algorithm engineering and experiments (ALENEX)*, pp. 81–89. SIAM, 2016.

Lin, Z., Wu, Y.-F., Peri, S. V., Jiang, J., and Ahn, S. Improving generative imagination in object-centric world models. In *International Conference on Machine Learning*, pp. 4114–4124, 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., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. Object-centric learning with slot attention, 2020.

Neal, R. M. Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265, 2000.

Nguyen-Phuoc, T., Richardt, C., Mai, L., Yang, Y.-L., and Mitra, N. Blockgan: Learning 3d object-aware scene representations from unlabelled images. *arXiv preprint arXiv:2002.08988*, 2020.

Niemeyer, M. and Geiger, A. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11453–11464, 2021.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation, 2021.

Reed, S., Akata, Z., Mohan, S., Tenka, S., Schiele, B., and Lee, H. Learning what and where to draw. In *NIPS'16 Proceedings of the 30th International Conference on Neural Information Processing Systems*, volume 29, pp. 217–225, 2016a.

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. Generative adversarial text to image synthesis. In *ICML'16 Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, pp. 1060–1069, 2016b.

Savarese, P., Kim, S. S. Y., Maire, M., Shakhnarovich, G., and McAllester, D. Information-theoretic segmentation by inpainting error maximization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4029–4039, 2021.

Sohn, K., Yan, X., and Lee, H. Learning structured output representation using deep conditional generative models. In *NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, volume 28, pp. 3483–3491, 2015.

Touvron, H., Cord, M., Matthijs, D., Massa, F., Sablayrolles, A., and Jegou, H. Training data-efficient image transformers distillation through attention. In *ICML 2021: 38th International Conference on Machine Learning*, pp. 10347–10357, 2021.

van den Oord, A., Vinyals, O., and koray kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6306–6315, 2017.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv: Learning*, 2018.

van Steenkiste, S., Kurach, K., Schmidhuber, J., and Gelly, S. Investigating object compositionality in generative adversarial networks. *Neural Networks*, 130:309–325, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

von Kügelgen, J., Ustyuzhaninov, I., Gehler, P. V., Bethge, M., and Schölkopf, B. Towards causal generative scene models via competition of experts. *arXiv preprint arXiv:2004.12906*, 2020.

Voynov, A., Morozov, S., and Babenko, A. Big gans are watching you: Towards unsupervised object segmentation with off-the-shelf generative models. In *arxiv:cs.LG*, 2021.

Watters, N., Matthey, L., Burgess, C. P., and Lerchner, A. Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes. *arXiv preprint arXiv:1901.07017*, 2019.

Wu, Y.-F., Yoon, J., and Ahn, S. Generative video transformer: Can objects be the words? In *ICML 2021: 38th International Conference on Machine Learning*, pp. 11307–11318, 2021.

Yan, X., Yang, J., Sohn, K., and Lee, H. Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision*, pp. 776–791, 2016.

Yang, C., Lamdouar, H., Lu, E., Zisserman, A., and Xie, W. Self-supervised video object segmentation by motion grouping. *arXiv: Computer Vision and Pattern Recognition*, 2021.

Yang, Y., Chen, Y., and Soatto, S. Learning to manipulate individual objects in an image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6558–6567, 2020.

Yuille, A. and Kersten, D. Vision as bayesian inference: analysis by synthesis? *Trends in cognitive sciences*, 10(7):301–308, 2006.

A SLATE

A.1 Architecture

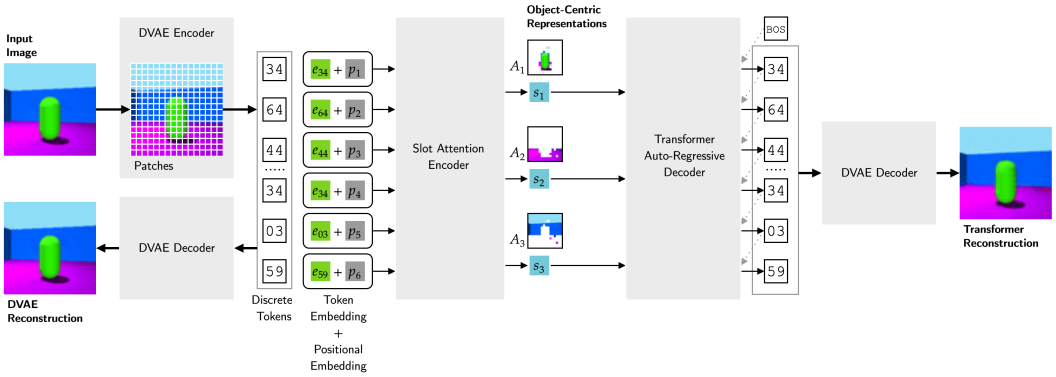


Figure 5: Architecture of SLATE. Our model receives an input image and we split it into patches. We encode each patch as a discrete token using a feedforward network (shown as the DVAE encoder). A DVAE decoder is trained to reconstruct each patch from its respective token using a simple MSE loss. We shall call this the DVAE reconstruction and this provides one of the two reconstruction pathways that we have in our architecture. For the second reconstruction pathway, the tokens obtained from the DVAE encoder are mapped to embeddings. To these embeddings, we add learned positional embedding to add the position information to the embedding of each token or the respective image patch. These resulting set of embeddings are provided as input to Slot Attention to discover N slots that summarize the contents of the discrete tokens and thereby summarizing the input image. The slots are then provided to a Transformer decoder which is trained to decode and reconstruct the discrete tokens of the input image using a cross-entropy loss. Given the slots, the tokens can be generated by the Transformer auto-regressively and then be decoded via the DVAE decoder to produce a reconstruction of the input image. This forms the slot-based reconstruction pathway and, unless specified, the term *reconstruction* would refer to this slot-based pathway.

A.2 Multi-headed Slot Attention

In Slot-Attention (Locatello et al., 2020), each slot undergoes several rounds of interactions with the input image in order to collect the contents for the slot representation. In each iteration, this interaction of a slot with the input image is performed using dot-product attention with slot as the query. The value returned by this attention is a simple weighted sum of the input cells being attended by the slot. This weighted sum as a pooling method and method of interaction between the attended cells can be insufficient for properly representing the attended cells. This is especially the case when the object has a complex shape and texture. To address this, we propose a multi-headed extension of slot attention. In this, we replace the dot-product attention with a multi-headed dot-product attention. By doing this, each slot attends to the different parts of the same object simultaneously and the result of the attention is concatenated. As this concatenated result passes through the RNN layers, these features for different parts of the same object can interact flexibly and result in a more rich encoding of the attended object. Recall that in standard slot attention, because there is only one attention head, different parts of the same object can only interact via the weighted sum which serves as a weak form of interaction as our results below shall indicate.

Implementation. We implement iterations of the Multi-headed Slot Attention as follows. For a given iteration, we first project the slots $\mathbf{S} = \mathbf{s}_{1:N}$ from the previous iteration into query vectors for each head. Similarly, we project the input cells $\mathbf{U} = \mathbf{u}_{1:T}$ into key and value vectors for each head.

$$\mathbf{Q}_m = \mathbf{W}_m^Q \mathbf{S},$$
$$\mathbf{K}_m = \mathbf{W}_m^K \mathbf{U},$$
$$\mathbf{V}_m = \mathbf{W}_m^V \mathbf{U}.$$

Next, we compute unnormalized attention proportions for each head m as by taking a dot-product as follows:

$$\frac{\mathbf{Q}_m \mathbf{K}_m^T}{\sqrt{D_K}}.$$

where D_K is the size of the key vectors. The attention map over the input cells $\mathbf{A}_{n,m}$ for slot n and head m is obtained by taking a softmax over both the N slots and the M heads.

$$[\mathbf{A}_{1:N,1}; \ldots; \mathbf{A}_{1:N,M}] = \text{softmax}(\big[\frac{\mathbf{Q}_1\mathbf{K}_1^T}{\sqrt{D_K}}; \ldots; \frac{\mathbf{Q}_M\mathbf{K}_M^T}{\sqrt{D_K}}\big], \texttt{dim} = \texttt{`slots'} \text{ and } \texttt{`heads'}).$$

We then obtain the attention result for a given slot n as follows.

$$\texttt{updates}_n = [\mathbf{A}_{n,1}\mathbf{V}_1; \ldots; \mathbf{A}_{n,M}\mathbf{V}_M]\mathbf{W}^O \tag{1}$$

where and $\mathbf{W}^O$ is an output projection matrix. With this update vector, the slot state is updated as follows as done in the standard Slot Attention (Locatello et al., 2020).

$$\mathbf{s}_n \leftarrow \text{GRU}(\texttt{updates}_n, \mathbf{s}_n),$$
$$\mathbf{s}_n \leftarrow \mathbf{s}_n + \text{MLP}(\text{LayerNorm}(\mathbf{s}_n)).$$

This results in the update slot that is given to the next iteration.

Results. We apply Multi-headed Slot Attention in the Bitmoji dataset for our model. We found significant improvement in the MSE when we used 4 heads as compared to the standard Slot Attention (equivalent to 1 head). This is reported in Table 2.

Model	MSE
Ours (`num_slot_heads=1`, `num_slots=8`)	391.15
Ours (`num_slot_heads=1`, `num_slots=15`)	371.05
Ours (`num_slot_heads=4`, `num_slots=8`)	**136.29**

Table 2: Effect of number of slots and number of slot heads on the reconstruction quality in Bitmoji dataset in SLATE. We compare the benefits of increasing the number of heads in our slot attention module and the effect of increasing the number of slots. We note that simply increasing the number of slots (from 8 to 15) does not improve the reconstruction quality significantly. However, increasing the number of heads (from 1 to 4) while keeping the number of slots same significantly improves the reconstruction quality.

We also noted that when using mixture decoder, using multi-headed slot attention does not help because the performance bottleneck is the Spatial Broadcast decoder (Watters et al., 2019) of the mixture decoder. This comparison is shown in Table 3.

Model	MSE
SA (`num_slot_heads=1`, `num_slots=8`)	394.5
SA (`num_slot_heads=4`, `num_slots=8`)	410.1

Table 3: Effect of number of slot heads on the reconstruction quality in Bitmoji dataset using Slot Attention (SA) with mixture-decoder. We note that with mixture decoder, having multiple heads does not benefit reconstruction as the performance bottleneck is the weak object component decoder based on Spatial Broadcast decoder (Watters et al., 2019)

Limitations. While multi-head attention helps information in the slots, in datasets with simple objects, having multiple heads can harm disentanglement of objects into different slots. This can occur because as each slots becomes more expressive, the network may be incentivised to collect information about multiple objects into the same slot. Therefore for our experiments on Shapestacks, we used regular slot attention module with 1 head. Because Bitmoji can have complex hair and face shapes, we used 4-headed slot attention module for this dataset in our model.

B Hyperparameters and Computational Requirements

We report the hyperparameters and the computational requirements for training our model in Table 4.

Dataset		Shapestacks	Bitmoji
Batch Size		50	50
LR Warmup Steps		30000	30000
Peak LR		0.0003	0.0001
Dropout		0.1	0.1
DVAE	Vocabulary Size	4096	4096
	Temp. Cooldown	1.0 to 0.1	1.0 to 0.1
	Temp. Cooldown Steps	30000	30000
	LR (no warmup)	0.0003	0.0003
Image Size		96	128
Image Tokens		576	1024
Transformer Decoder Specifications	Layers	8	8
	Heads	8	8
	Hidden Dim.	192	192
Slot Attention Specifications	Slots	12	8
	Iterations	7	3
	Slot Dim.	192	192
Training Cost	GPU Usage	14GB	64GB
	Days	5 Days	3.5 Days

Table 4: Hyperparameters used for our model and computation requirements for each dataset.

Dropout. We found it beneficial to perform dropout during training of our model. We apply attention dropout of 0.1 in the transformer decoder. We also apply the same dropout of 0.1 after positional embedding is added to the patch token embedding i.e.

$$\mathbf{u}_i \leftarrow \text{Dropout}(\text{Dictionary}_\phi(\mathbf{z}_i) + \mathbf{p}_{\phi,i}, \;\; \texttt{dropout=}0.1).$$

Learning Rate Schedules. For training the parameters of DVAE, we noted that a constant learning rate of 3e-4 produced good discretization of patches and low reconstruction error of the patches decoded from the DVAE code. Smaller learning rates lead to poorer use of the dictionary and poorer reconstruction of patches.

For the weights of slot attention encoder, the transformer and the learned embeddings, we found a learning rate warm-up schedule helpful. For this warm-up, we increase the learning rate linearly from 0.0 to the peak learning rate in the first 30K training steps. After this, at the end of every epoch, we monitor the loss on the validation set. If the validation loss does not decrease for 8 consecutive epochs, we reduce the learning rate by a factor of 1/2.

C Additional Qualitative Results

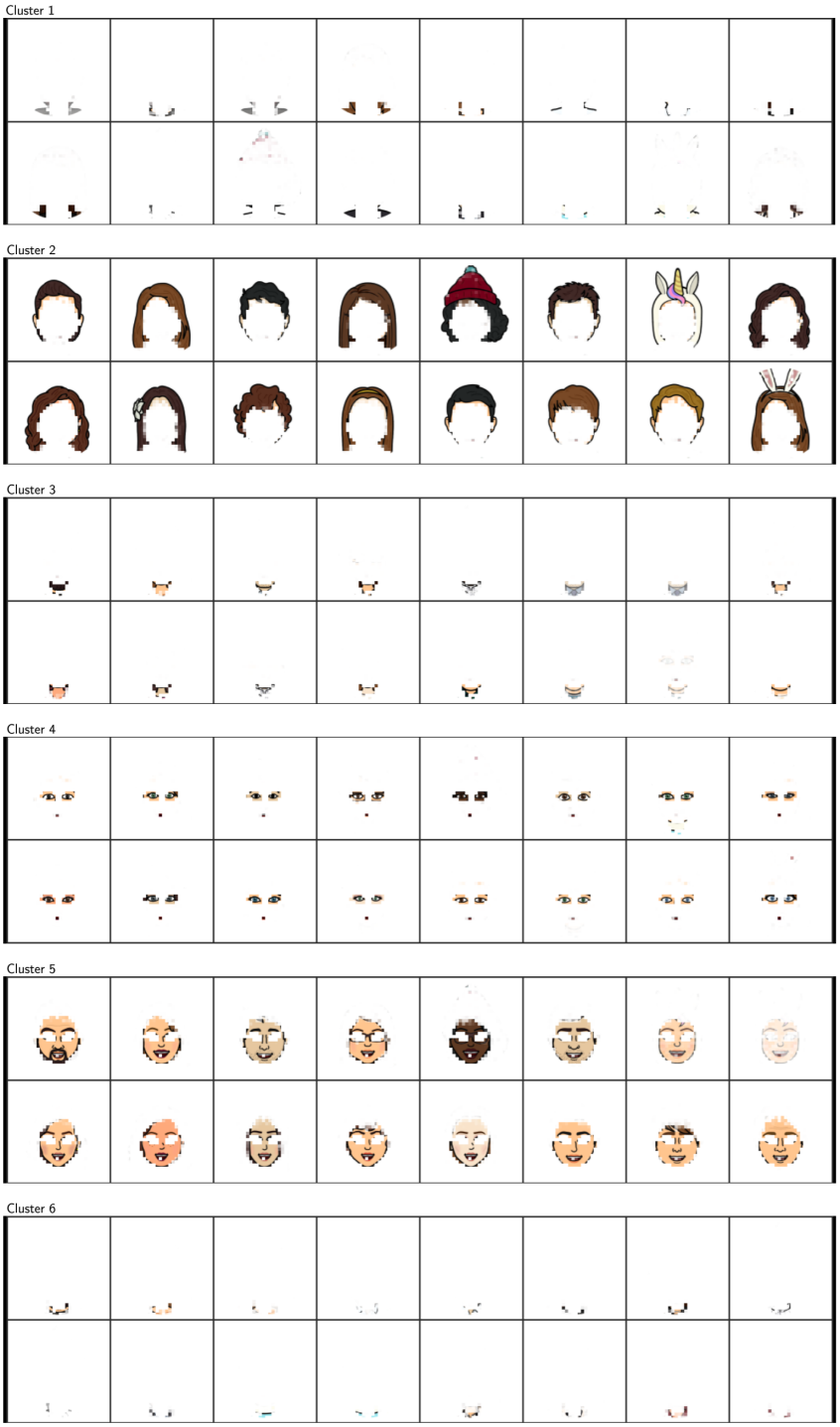


Figure 6: Clusters in the concept library obtained by applying K-means on slots obtained from the Bitmoji dataset.

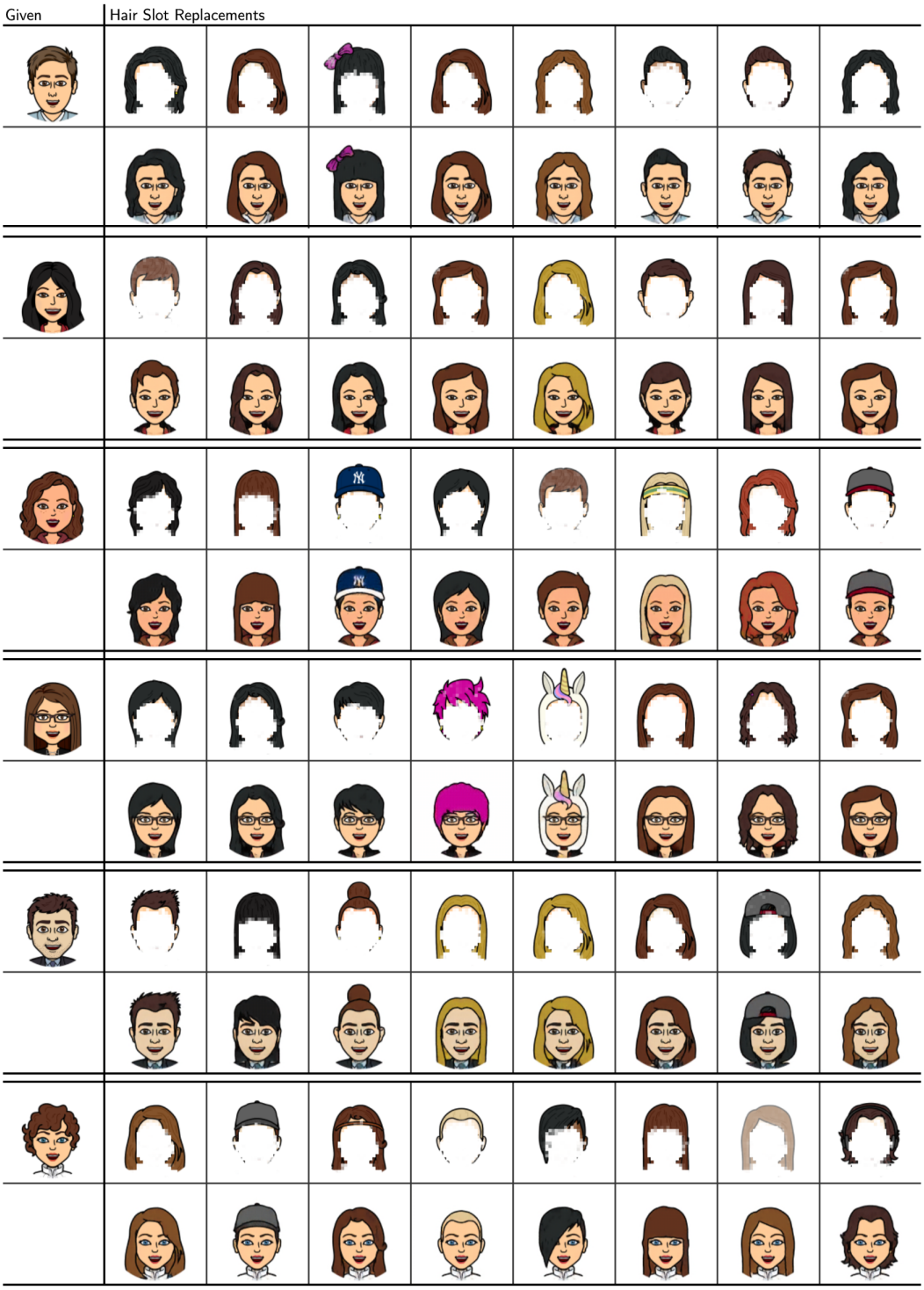


Figure 7: Qualitative Results for Hair Slot Replacements in Bitmoji dataset. For each image, the first row shows the hair slot that will be used as a replacement and the second row shows the generation from our model after the hair slot is replaced.

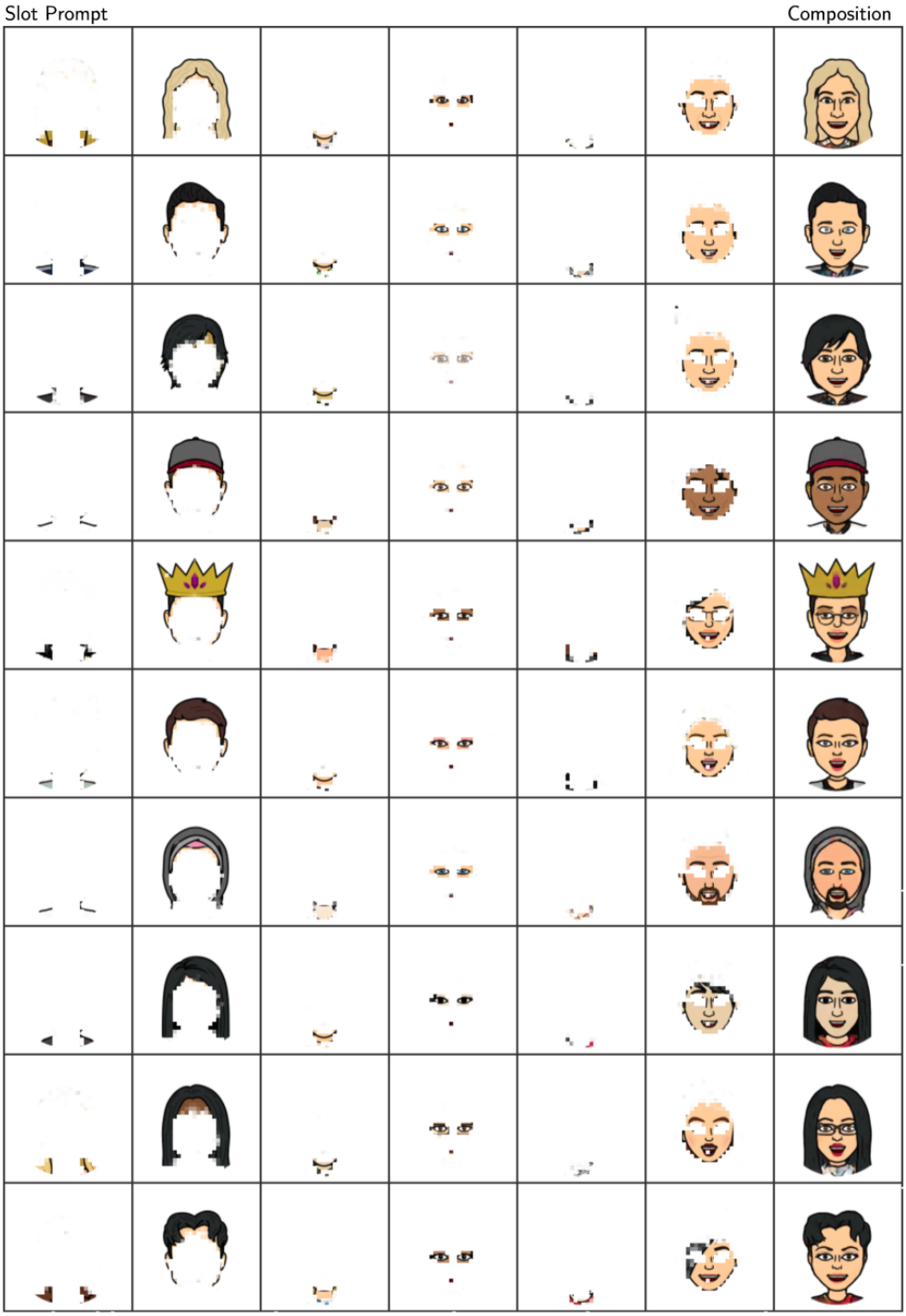


Figure 8: Qualitative Results for Compositional Generation in Bitmoji dataset using our model.

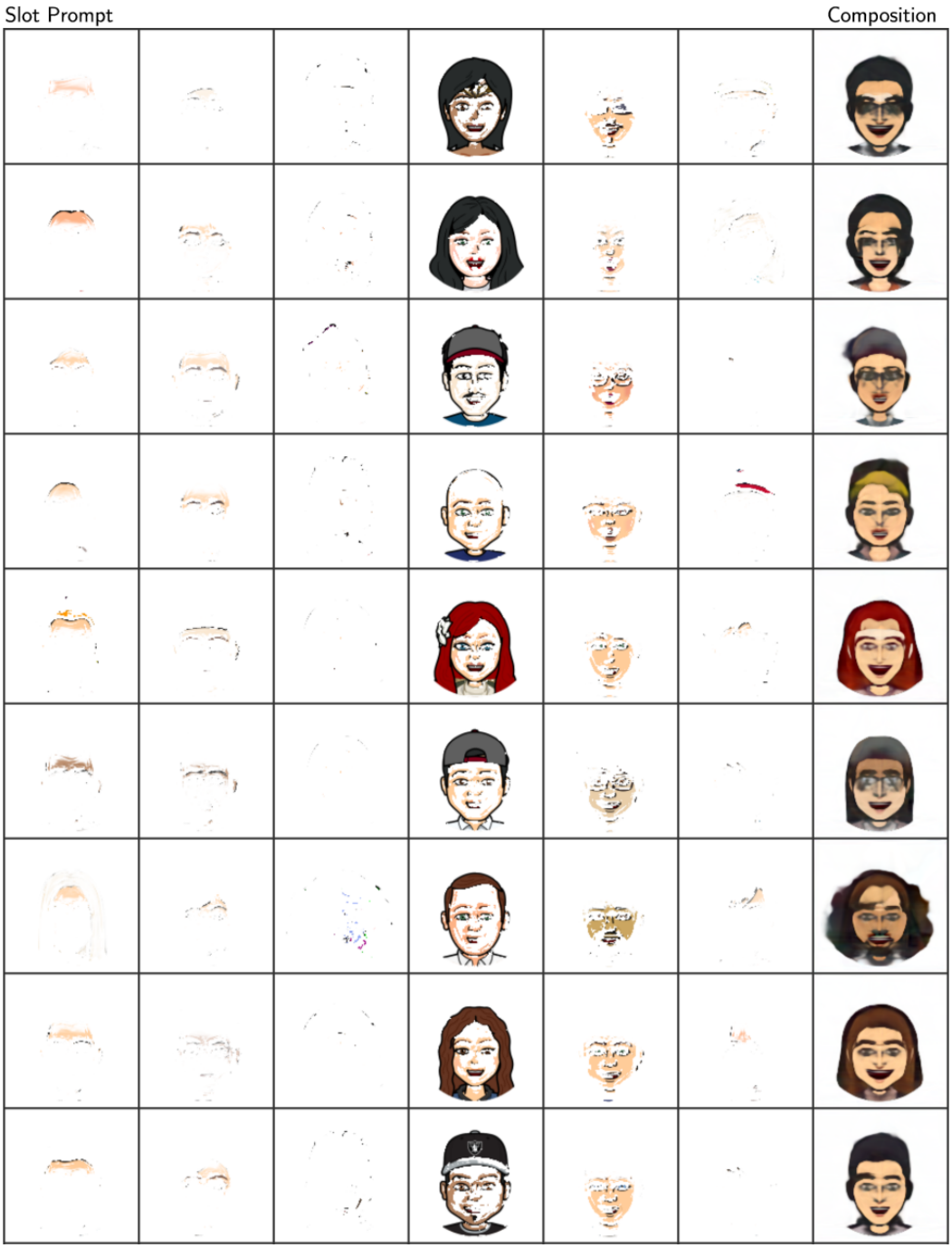


Figure 9: Qualitative Results for Compositional Generation in Bitmoji dataset using Slot Attention.

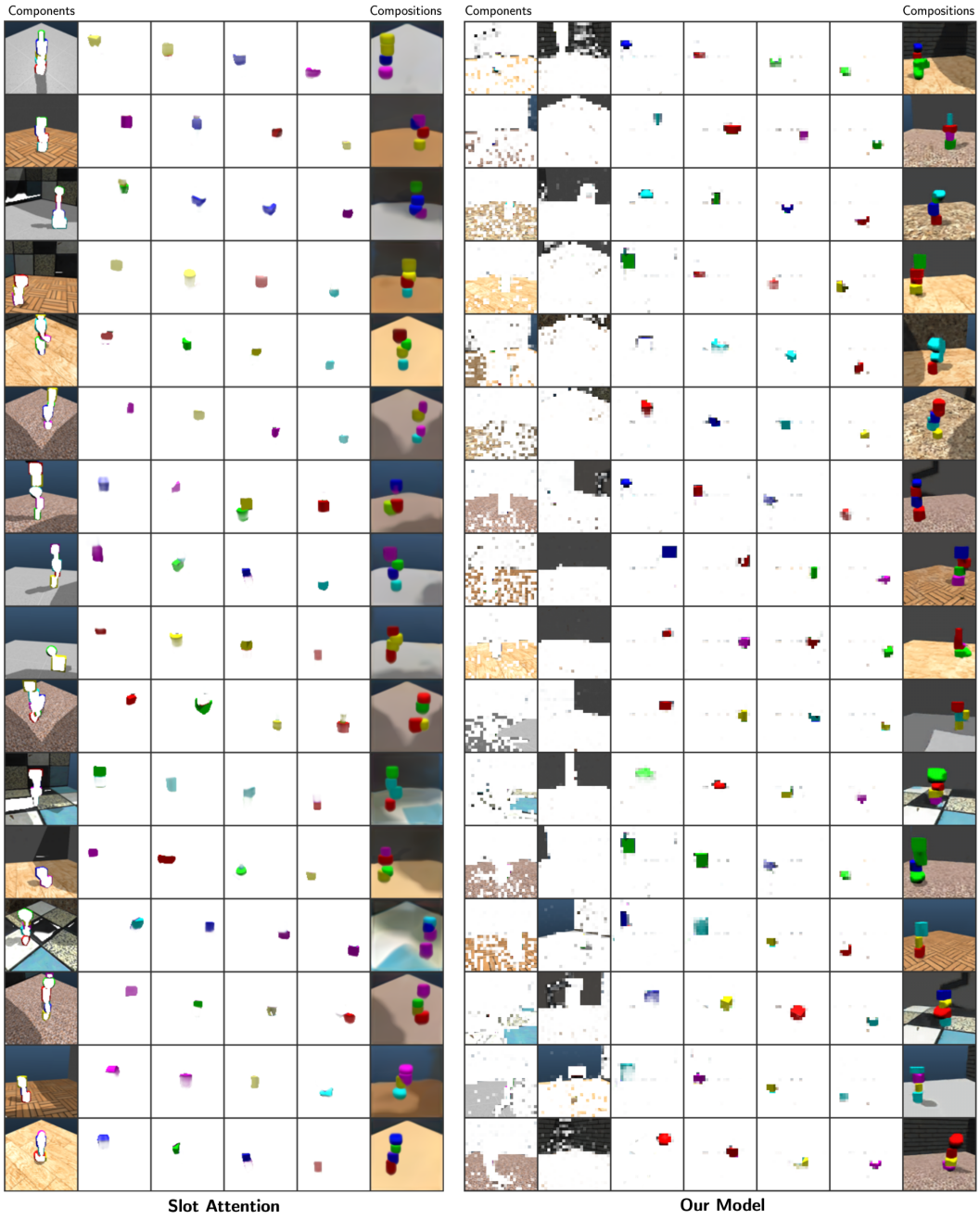


Figure 10: Qualitative Comparison of Compositional Generation in Shapestacks.

D Discussion

D.1 Strong Decoder in Mixture Likelihood Models Make Slots Capture Multiple Objects

One of the attractive properties of our model is that our Transformer decoder produces images with significantly better detail than Slot Attention. Slot Attention fails to render such details largely because of Spatial Broadcast Decoder that is used to render each object component. Spatial Broadcast Decoder decodes images by mapping a pixel coordinate and the provided slot vector to RGB pixel values and a mask using an MLP. This MLP prefers to learn simple mappings from pixel coordinate to RGB values. At the start of the training, this function tends to be a constant or a linear function of pixel coordinate and only after more training does this function achieve a more complex input-output behavior.

Hence, when a Spatial Broadcast Decoder is used, the decoder of each object component is biased in early training to prefer modeling segments without much RGB variation within the object patch. This property is important to guide the training of Slot Attention encoder to learn to disentangle separate objects and it has been exploited in several prior works on unsupervised object-centric representation learning. To understand why this occurs, consider a grayscale image that contains two balls, one colored white with pixel value 1.0 and the other colored grey with pixel value 0.5. Now consider the case when each slot represents one object in the scene. In this case, the decoder for the white ball simply needs to output a value 1.0 for all pixel coordinates and this is a constant function. Similarly, for the grey ball, the decoder needs to output a value 0.5 for all pixel coordinates. In contrast, consider the case where a single slot represents both the white and the grey balls. In this case, the decoder needs to map the pixels coordinates for the white ball to 1.0 and the pixel coordinates for the grey ball to 0.5 within the same function. This function is not longer a constant and is therefore much more complex than the functions that arose when each slot modeled separate objects. Furthermore, as objects might be positioned randomly, this mapping needs to quickly change when the positions of the balls change in the scene depending on information from the slot representation. This makes the mapping even more complex when more than one objects are modeled by the same slot. Hence, the network naturally prefers to model and decode different objects via different slots. As a result, the encoder is incentivized to cluster the pixels of the input image into objects. While this property could be utilized in simple datasets which have objects without much RGB variation and detail within the object, it can fail to render images properly in datasets such as Shapestacks with floor with complex textures or Bitmoji with complex face details and the shape of the hair.

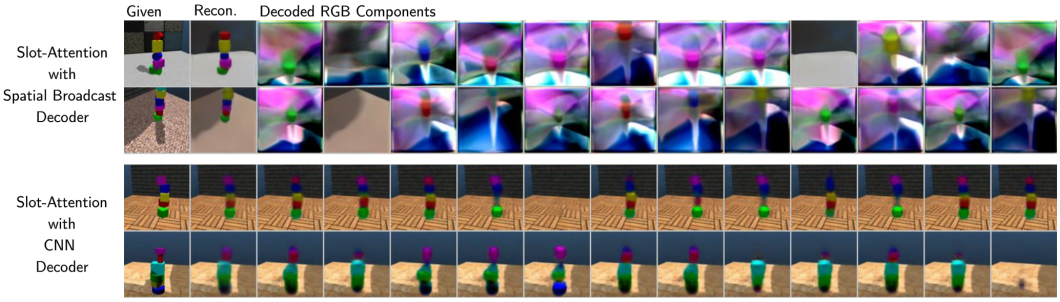


Figure 11: Analysis of decoder capacity in Slot Attention when using a mixture-based likelihood model for decoding. We note that when we use a powerful CNN decoder to decode the RGB components and decoding masks for the objects, then a single component tries to model all the objects in the scene instead of each slot representing a single object as in the case of Spatial Broadcast decoder.

While one may argue that this could be fixed simply by replacing the Spatial Broadcast Decoder with a more powerful decoder for the object components. However, because weak decoder was encouraging the disentanglement of objects as argued above, hence by the same argument, when a powerful decoder is used, a single slot no longer has the incentive to model one object. Thus each slot can try to represent and model more than one object in the scene. We tested this empirically by replacing the Spatial Broadcast Decoder of Slot Attention with a powerful CNN decoder. We found that this causes the slots to capture multiple objects and in some cases a single slot may try to model all the contents of the scene as a single object segment. This is shown in Figure 11.

Figure 12: Reconstruction and attention maps of slots in SLATE given the input images for Shapestacks. We note that attention maps effectively localize on the individual blocks.

Hence for unsupervised object discovery, this historically presented a trade-off between decoding capacity and the ability of the model to disentangle the objects. In contrast, our work shows that such a trade-off can be eliminated if Transformer is used as the decoder. Our model can not only detect objects without supervision but also render their fine details during decoding. See Figures 12 and 13.

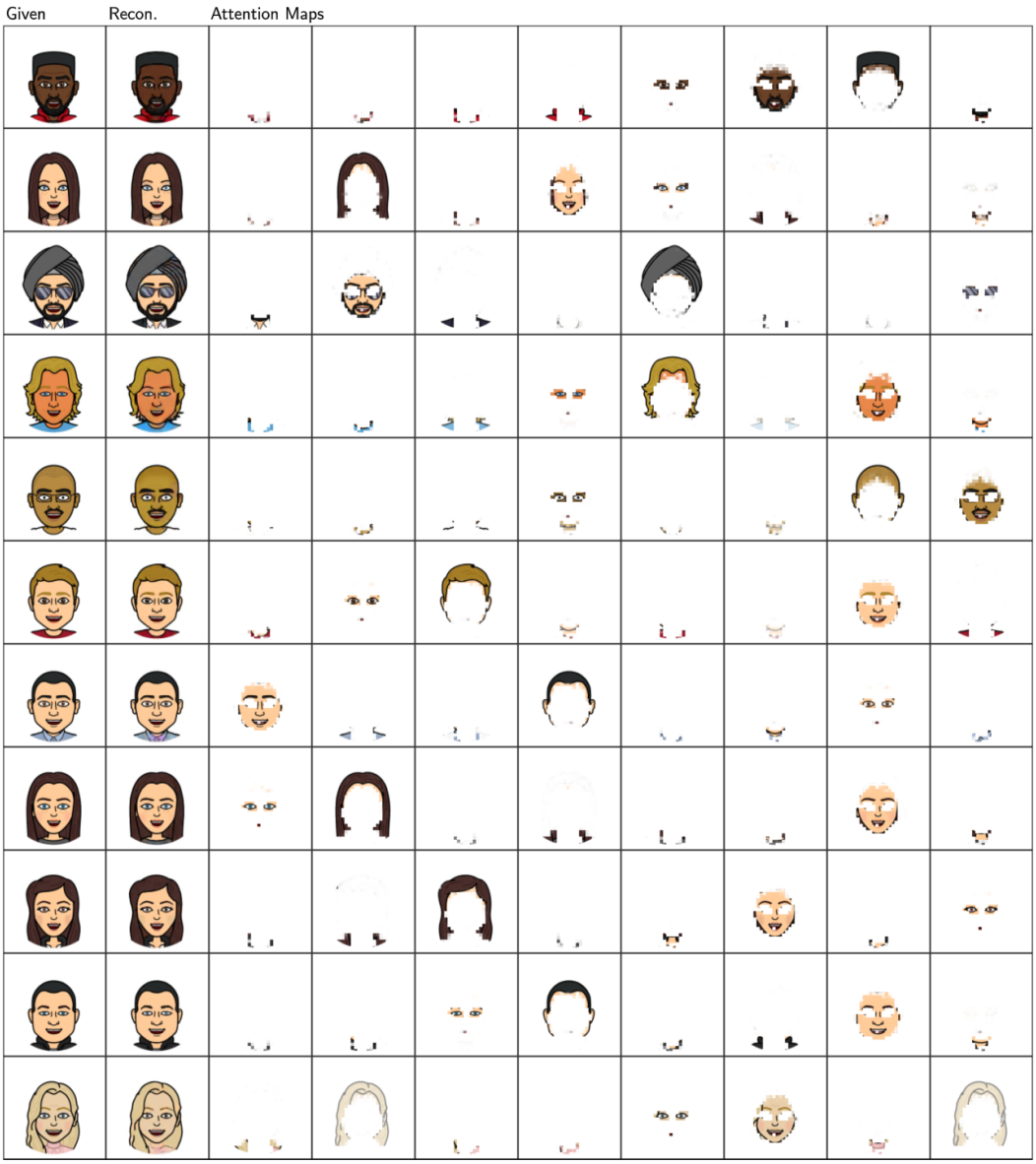


Figure 13: Reconstruction and attention maps of slots in SLATE given the input images for Bitmoji. We note that attention maps effectively localize on the individual meaningful segments of the faces.