SIMONe: View-Invariant, Temporally-Abstracted Object Representations via Unsupervised Video Decomposition

Anonymous Author(s)
Affiliation
Address
email

Abstract

To help agents reason about scenes in terms of their building blocks, we wish to extract the compositional structure of any given scene (in particular, the configuration and characteristics of objects comprising the scene). This problem is particularly difficult when scene structure needs to be inferred while also estimating the agent's location/viewpoint, as the two variables jointly give rise to the agent's observations. We present an unsupervised variational approach to this problem. Leveraging the shared structure that exists across different scenes, our model learns to infer two sets of latent representations from RGB video input: a set of "object" latents, corresponding to the time-invariant, object-level contents of the scene, as well as a set of "frame" latents, corresponding to global time-varying elements such as viewpoint. This factorization of latents allows our model, SIMONe to represent object attributes in an allocentric manner which does not depend on viewpoint. Moreover, it allows us to disentangle object dynamics and summarize their trajectories as time-abstracted, view-invariant, per-object properties. We demonstrate these capabilities, as well as the model's performance in terms of view synthesis and instance segmentation, across three procedurally generated video datasets.

1 Introduction

The problem of *unsupervised visual scene understanding* has become an increasingly central topic in machine learning [1, 2]. The attention is merited by potential gains to reasoning, autonomous navigation, and myriad tasks. However, within the current literature, different studies frame the problem in different ways. One approach aims to decompose images into component objects and object features, supporting (among other things) generation of alternative data that permits insertion, deletion, or repositioning of individual objects [3–6]. Another approach aims at a very different form of decomposition—between allocentric scene structure and a variable viewpoint—supporting generation of views of a scene from new vantage points [7–9] and, if not supplied as input, estimation of camera pose [10]. Although there is work addressing both of these approaches concurrently in the supervised setting [11], very few previous studies have approached the combined challenge in the unsupervised case. In this work, we introduce the Sequence-Integrating Multi-Object Net (SIMONe), a model which pursues that goal of tackling object-level and viewpoint-level scene decomposition and synthesis without supervision. SIMONe is designed to handle these challenges without privileged information concerning camera pose, and works in dynamic scenes.

Given a video of a scene our model is able to decouple scene structure from viewpoint information (see Figure 1). To do so, it utilizes video-based cues, and a structured latent space which separates time-invariant per-object features from time-varying global features. These features are inferred using a transformer-based network which integrates information jointly across space and time.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

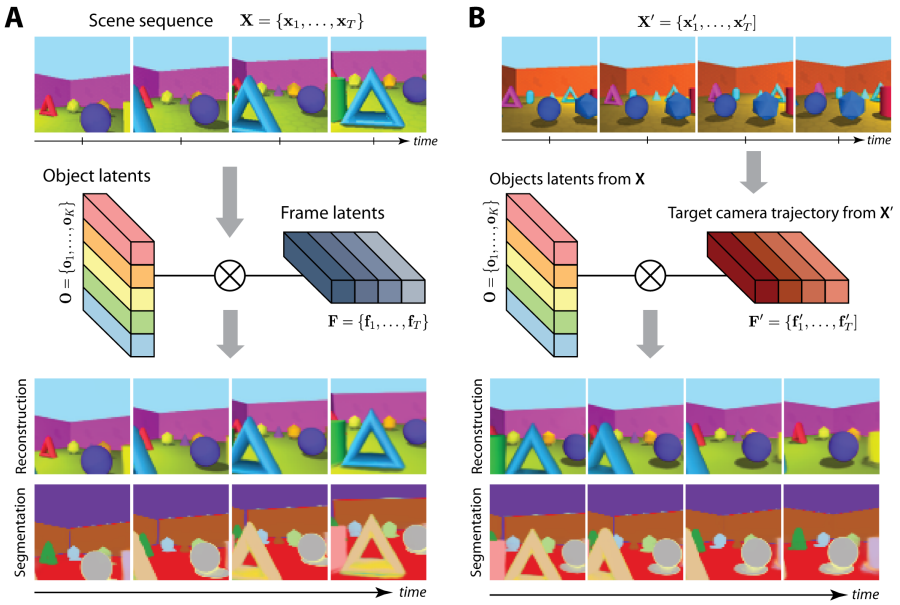

Figure 1: **A.** SIMONe decomposes and factorises a scene sequence $\mathbf{X}$ into *scene content* ("object latents", constant across the sequence) and *global/view content* ("frame latents", one per frame) without supervision. Its spatio-temporal attention-based inference naturally allows stable object tracking (e.g. the blue triangle is assigned the same segment across frames). **B.** Object latents of a given sequence $\mathbf{X}$ can be re-composed with the frame latents of a different (i.i.d.) sequence $\mathbf{X}'$ to generate a consistent rendering of the same scene content (i.e. objects and their properties, relative arrangements, and segmentation assignments) from entirely different viewpoints. Notice that both camera pose and lighting are transferred, as evidenced by the wall corners in the background and the shadow below the purple ball. See our website for more, animated examples.

Second, our method seeks to summarize object trajectories in time-invariant object codes. SIMONe learns to disentangle not only static object attributes (and their 2D spatial masks), but also object trajectories, without any prior notion of these objects, from videos alone. The learnt trajectory features are temporally abstract and per object; they are captured independently of the dynamics of camera pose, which being a global property, is captured in the model's per-frame (time-varying) latents.[1]

Our model thus advances the state of the art in unsupervised, object-centric scene understanding by satisfying the following desiderata: **(1)** decomposition of multi-object scenes from RGB videos alone; **(2)** handling of changing camera pose, and simultaneous inference of scene contents and viewpoint from correlated views (i.e. sequential observations of a moving agent); **(3)** learning of structure across diverse scene instances (i.e. procedurally sampled contents); **(4)** object representations which summarize static object attributes like color or shape, view-dissociated properties like position or size, as well as time-abstracted trajectory features like direction of motion; **(5)** no explicit assumptions of 3D geometry, no explicit dynamics model, no specialized renderer, and few a priori modeling assumptions about the objects being studied; and **(6)** simple, scalable modules (for inference and rendering) to enable large-scale use.

2 Related Work

Given the multifaceted problem it tackles, SIMONe connects across several areas of prior work. We describe its nearest neighbors from three scene understanding domains below:

Scene decomposition models. **(1)** Our work builds on a recent surge of interest in unsupervised scene decomposition and understanding, especially using slot structure to capture the objects in a scene [12]. One line of work closely related to SIMONe includes methods like [3–6, 13], which all share SIMONe's Gaussian mixture pixel likelihood model. While these prior methods handled

[1] Animated figures are at https://sites.google.com/view/simone-neurips/.

only static scenes, more recent work [14–16] has extended them to videos with promising results. Nevertheless, these approaches have no mechanism or inductive bias to separate view information from scene contents. Moreover, many of them are conditioned on extra inputs like the actions of an agent/camera to simplify inference. **(2)** Another family of decomposition models originated with Attend, Infer, Repeat (AIR) [17]. AIR's recurrent attention mechanism does split images into components with separate appearance and pose latents each. Later work [6, 18–21] also extended the model to videos. Despite their structured latents, these models do not learn to distill object appearance into a time-invariant representation (as their appearance and pose latents are free to vary as a function of time). They also require separate object discovery and propagation modules to handle appearing/disappearing objects. In contrast, SIMONe processes a full sequence of images using spatio-temporal attention and produces a single time-invariant latent for each object, hence requiring no explicit transition model or discovery/propagation modules.

Multi-view scene rendering models. Models which assume viewpoint information for each image like GQN [7], SRNs [8], and NeRF [9] have shown impressive success at learning implicit scene representations and generating novel views from different viewpoints. Recent work [22–25] has further extended these models to videos using deformation fields to model changes in scene geometry over time. In contrast to SIMONe, these models can achieve photorealistic reconstructions by assuming camera parameters and viewpoint information. To allow a direct comparison, we use a view-supervised version of SIMONe in Section 4.1. There is also recent work [26, 27] that relaxes the known viewpoint constraint; but these still model single scenes at a time, which prevents them from exploiting regularities over multiple scenes. A more recent line of work [28–30] explored amortizing inference in these models by learning to map from a given set of images to scene latents; however, these models are not temporal by design and cannot yet handle videos. Note that all of these models treat the whole scene as a single entity and do not decompose the scene into objects. One exception here is [31], which represents objects with separate pose and appearance latents. However, this model is purely generative and cannot infer object latents from a given scene like SIMONe does.

Simultaneous localization and mapping. The problem of inferring scene representations in a novel environment by exploring it (rather than assuming given views and viewpoint information) is well studied in robotics and vision [10]. Classic SLAM techniques often rely on EM [32, 33] or particle filters [34] to infer viewpoint and scene contents jointly. While our problem is slightly simpler (we can leverage shared structure across scene instances; certain elements such as the layout of the room are held constant; and we use offline data rather than active exploration), our approach of using a factorized variational posterior provides a learning-based solution to the same computational problem. Our simplified setting is perhaps justified by our unsupervised take on the problem. On the other hand, we don't assume simplifications which may be common in robotics practice (e.g. known camera properties like field of view; or the use of multiple cameras or depth measurement). Most popular SLAM benchmarks [35–37] are on unstructured 3D scenes and hence it was not straightforward for us to compare directly to classic methods. But an encouraging point of overlap is that object-centric SLAM formulations [11] as well as learning-based solutions [38, 39] are active topics of research. Our work could open new avenues in object-centric scene mapping without supervision.

3 Model

SIMONe is a variational auto-encoder [40] consisting of an inference network (encoder) which infers latent variables from a given input sequence and a generative process (decoder) which decodes these latents back into pixels. Using the Evidence Lower Bound (ELBO), the model is trained to minimize a pixel reconstruction loss and latent compression KL loss. Crucially, SIMONe relies on a factorized latent space which enforces a separation of static object attributes from global, dynamic properties such as camera pose. We introduce our latent factorization and generative process in Section 3.1. Then in Section 3.2, we describe how the latents can be inferred using a transformer-based encoder, significantly simplifying the (recurrent or autoregressive) architectures used in prior work. Finally, we fully specify the training scheme in Section 3.3.

3.1 Latent Structure and Generative Process

Our model aims to capture the structure of a scene, observed as a sequence of images from multiple viewpoints (often along a smooth trajectory, though this is not a requirement). Like many recently

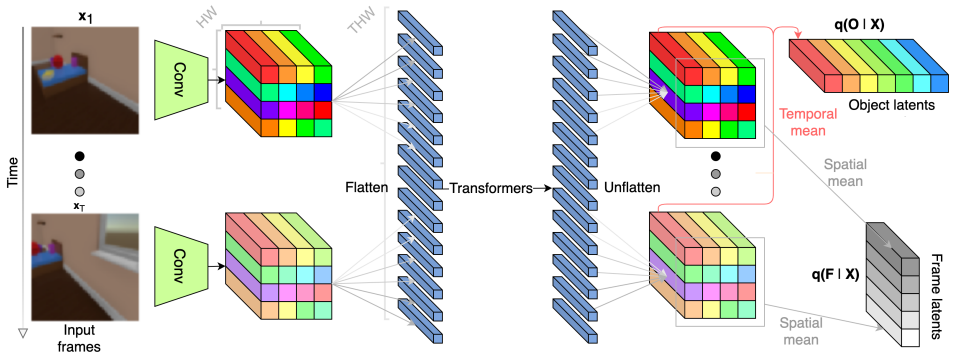

Figure 2: **Architecture** of the SIMONe inference network $\mathcal{E}_\phi$. The transformers integrate information jointly across space and time to infer the object and frame posterior parameters.

proposed object-centric models we choose to represent the scene as a set of K *object* latent variables $\mathbf{O} \coloneqq \{\mathbf{o}_k\}_{k=1}^K$. These are invariant by construction across all frames in the sequence (i.e. their distribution is constant through time, and expected to summarize information across the whole sequence). We also introduce T *frame* latents $\mathbf{F} \coloneqq \{\mathbf{f}_t\}_{t=1}^T$, one for each frame in the sequence, which capture time-varying information. Note that by choosing this factorization we reduce the number of required latent variables from $K \cdot T$ to $K + T$. The latent prior $p(\mathbf{O}, \mathbf{F}) = \prod_k \mathcal{N}(\mathbf{o}_k \mid \mathbf{0}, \mathbf{I}) \prod_t \mathcal{N}(\mathbf{f}_t \mid \mathbf{0}, \mathbf{I})$ is a unit spherical Gaussian, assuming and enforcing independence between object latents, frame latents, and feature dimensions.

We model each pixel as a Gaussian mixture model with K components. Given the latents, we assume all pixels and all frames to be independent. Therefore, the mixture weights for any RGB pixel $\mathbf{x}_{t,i}$ capture which object k "explains" that pixel ($1 \leq i \leq HW$). The corresponding mixing logits $\hat{m}_{k,t,i}$ and pixel (reconstruction) means $\boldsymbol{\mu}_{k,t,i}$ are computed for every slot k at a specific time-step t and specific pixel location $\mathbf{l}_i$ using a decoder $\mathcal{D}_\theta$:

$$\hat{m}_{k,t,i}, \boldsymbol{\mu}_{k,t,i} = \mathcal{D}_\theta(\mathbf{o}_k, \mathbf{f}_t; \mathbf{l}_i, t) \tag{1}$$

$$p(\mathbf{x}_{t,i} \mid \mathbf{o}_1, ..., \mathbf{o}_K, \mathbf{f}_t; t, \mathbf{l}_i) = \sum_k m_{k,t,i} \mathcal{N}(\mathbf{x}_{t,i} \mid \boldsymbol{\mu}_{k,t,i}; \sigma_x) \tag{2}$$

We decode each pixel independently, "querying" our *pixel-wise* decoder using the sampled latents, coordinates $\mathbf{l}_i \in [-1, 1]^2$ of the pixel, and time-step being decoded as inputs. The decoder's architecture is an MLP or 1x1 CNN. (See Appendix A.3.1 for the exact parameterization as well as an overview of the generative process). By constraining the decoder to work on individual pixels, we can save on memory and computation, using a subset of pixels as training targets (as opposed to full images; this is elaborated in Section 3.3). Once they are decoded, we obtain the mixing weights $m_{k,t,i}$ by taking the softmax of the mask logits across the K slots: $m_{k,t,i} = \texttt{softmax}_k(\hat{m}_{k,t,i})$. Equation 2 specifies the full pixel likelihood, where σ_x is a scalar hyperparameter.

3.2 Inference

Given a sequence of frames $\mathbf{X} \coloneqq \{\mathbf{x}_t\}_{t=1}^T$ we now wish to infer the corresponding slot latents $\mathbf{O}$ and frame latents $\mathbf{F}$. The exact posterior distribution $p(\mathbf{O}, \mathbf{F} \mid \mathbf{X})$ is intractable so we resort to using a Gaussian approximate posterior $q(\mathbf{O}, \mathbf{F} \mid \mathbf{X})$. The approximate posterior is parameterized as the output of an inference (encoder) network $\mathcal{E}_\phi(\mathbf{X})$ which outputs the mean and (diagonal) log-scale for all latent variables given the input sequence.

SIMONe's inference network is based on the principle that spatio-temporal data can be processed *jointly* across space and time using transformers. Beyond an initial step, we don't need the translation invariance of a CNN, which forces spatial features to interact gradually via a widening receptive field. Nor do we need the temporal invariance of an RNN which forces sequential processing. Instead, we let feature maps interact simultaneously across the cross-product of space and time. See Figure 2 for an overview of our encoder architecture implementing this.

Concretely, each frame $\mathbf{x}_t$ in the sequence is passed through a CNN which outputs HW spatial feature maps at each time-step (containing C channels each). HW can be larger than the number of

object latents K. (For all results in the paper, we set H and W to 8 each, and $K = 16$.) The rest of the inference network consists of two transformers $\mathcal{T}_1$ and $\mathcal{T}_2$. $\mathcal{T}_1$ takes in all THW feature maps. Each feature map attends to all others as described. $\mathcal{T}_1$ outputs THW transformed feature maps. When $HW > K$, we apply a spatial pool to reduce the number of slots to TK (see Appendix for more details). These slots serve as the input to $\mathcal{T}_2$, which produces an equal number of output slots. Both transformers use absolute (rather than relative) positional embeddings, but these are 3D to denote the spatio-temporal position of each slot. We denote the output of $\mathcal{T}_2$ as $\hat{\mathbf{e}}_{t,k}$. This intermediate output is aggregated along separate axes (and passed through MLPs) to obtain T frame and K object posterior parameters respectively. Specifically, $\boldsymbol{\lambda}_{\mathbf{o}_k} = \text{mlp}_o(1/T \sum_t \hat{\mathbf{e}}_{t,k})$ while $\boldsymbol{\lambda}_{\mathbf{f}_t} = \text{mlp}_f(1/K \sum_k \hat{\mathbf{e}}_{t,k})$. Using these posterior parameters we can sample the object latents $\mathbf{o}_k \sim \mathcal{N}(\boldsymbol{\lambda}^{\mu}_{\mathbf{o}_k}, \exp(\boldsymbol{\lambda}^{\sigma}_{\mathbf{o}_k})\mathbb{1})$, and the frame latents $\mathbf{f}_t \sim \mathcal{N}(\boldsymbol{\lambda}^{\mu}_{\mathbf{f}_t}, \exp(\boldsymbol{\lambda}^{\sigma}_{\mathbf{f}_t})\mathbb{1})$.

3.3 Loss and Training

The model is trained end to end by minimizing the following negative ELBO:

$$\frac{-\alpha}{T_d H_d W_d} \sum_{t=1}^{T_d} \sum_{i=1}^{H_d W_d} \log p(\mathbf{x}_{t,i} \mid \mathbf{o}_1, ..., \mathbf{o}_K, \mathbf{f}_t; t, \mathbf{l}_i) + \frac{\beta_o}{K} \sum_k D_{KL}(q(\mathbf{o}_k \mid \mathbf{X}) \parallel p(\mathbf{o}_k)) + \frac{\beta_f}{T} \sum_t D_{KL}(q(\mathbf{f}_t \mid \mathbf{X}) \parallel p(\mathbf{f}_t)) \tag{3}$$

We normalize the data log-likelihood by the number of decoded pixels ($T_d H_d W_d$) to allow for decoding fewer than all input pixels (THW). This helps scale the size of the decoder (without reducing the learning signal, due to the correlations prevalent between adjacent pixels). Normalizing by $1/T_d H_d W_d$ ensures consistent learning dynamics regardless of the choice of how many pixels are decoded. α is generally set to 1, but available to tweak in case the scale of β_o and β_f is too small to be numerically stable. Unless explicitly mentioned, we set $\beta_o = \beta_f$. See Appendix A.3 for details.

4 Comparative Evaluation

In order to evaluate the model we focus on two tasks: novel view synthesis and video instance segmentation. On the first task (Section 4.1), we highlight the benefit of view information when it is provided as ground truth to a simplified version of our model (denoted "SIMONe-VS" for view supervised), as well as baseline models like GQN [7] and NeRF-VAE [41] which both use view information in their pipeline. On the second task (Section 4.2), we deploy the fully unsupervised version of our model; we showcase not only the possibility of inferring viewpoint from data, but also its benefit to extracting object-level structure in comparison to methods like MONet [3], Slot Attention [13], and Sequential IODINE [4].

We demonstrate our results on three procedurally generated video datasets of multi-object scenes. In increasing order of difficulty, they are: **Objects Room 9**, **CATER** (moving camera) [42], and **Playroom** [43]. These were chosen to meet a number of criteria: we want at least 9-10 objects per scene (there could be fewer in view, or as many as 25 in the case of Playroom). We want a moving camera, ideally with a randomized initial position (the only exception is CATER, where the camera moves rapidly but is initialized at a fixed position to help localization). We want ground-truth object masks to evaluate our results quantitatively. We also want richness in terms of lighting, texture, object attributes, and other procedurally generated elements. Finally, we want independently moving objects in at least one dataset (to evaluate trajectory disentangling and temporal abstraction), and unpredictable camera trajectories in another (the Playroom dataset is sampled using an arbitrary agent policy, so the agent is not always moving). Details on all datasets are in Appendix A.2.

4.1 View synthesis (with viewpoint supervision)

We first motivate view-invariant object representations by considering the case when ground-truth camera pose *is* provided to our model (a simplified variant we call "SIMONe-VS"). In this scenario, we don't infer any frame latents. Rather the encoder and decoder are conditioned on the viewpoint directly. This *view-supervised* setting is similar to models like GQN and NeRF which represent the contents of a scene implicitly and can be queried in different directions.

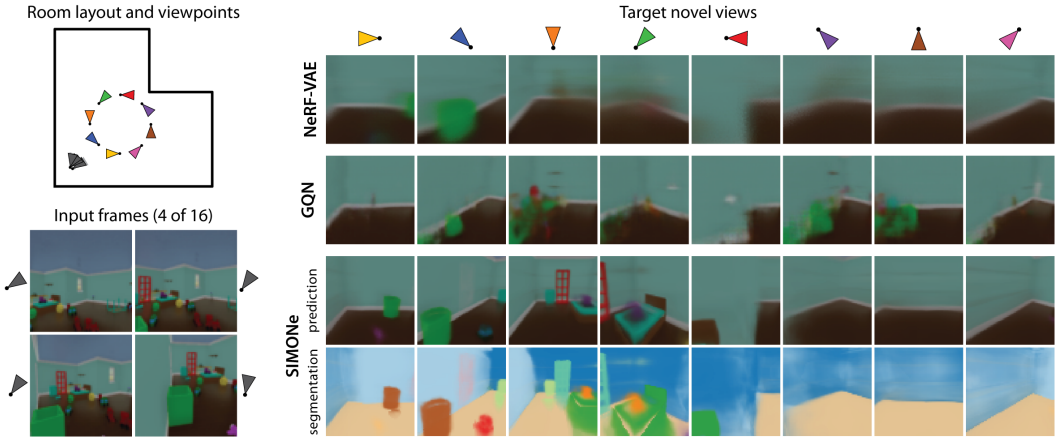

Figure 3: **Comparison of scene representation and view interpolation capabilities** between SIMONe-VS, NeRF-VAE, and GQN. All models partially observe a procedurally generated Playroom from a given sequence of frames (we visualize 4 of the 16 input frames fed to the models). Then, we decode novel views on a circular trajectory around the room, with the yaw linearly spaced in $[-\pi, \pi]$. NeRF-VAE retains very little object structure, while GQN hallucinates content. SIMONe-VS can produce fine reconstructions of objects that it observes even partially or at a distance (such as the bed or shelves in the scene). SIMONe-VS also segments the scene as a bonus.

We compare the decoded reconstructions of three such models on different views in the Playroom. The models are provided the same set of 16 initial context frames, partially revealing a sampled room. Having observed many instances of the Playroom, the models are tasked with extrapolating to unobserved views of the revealed scene (without being retrained). Note that the L-shaped layout of the room is shared across different scenes. The position, size, and choice of all objects, furniture, doors, shelves, etc is procedurally generated in different scenes. The floor, skirting, doors, and windows remain the same color but other object and background colors are sampled. NeRF itself is not directly suitable for this task as it needs to be retrained on each scene, whereas we want our model to learn common structure across diverse settings and infer the specific of a given room at evaluation time. NeRF-VAE addresses this issue and makes it directly comparable to our model.

To set up the comparison, we first trained SIMONe-VS and evaluated its log-likelihood on Playroom sequences. Then, we trained GQN and NeRF-VAE using constrained optimization (GECO [44]) to achieve roughly the same log likelihood per pixel. This allows us to compare the cost of representing scenes (i.e. the KL loss) across the different models, which speaks to the effectiveness of their latent structure. A priori, we expect an object-centric model to be able to represent multi-object scenes (controlling for reconstruction fidelity) with lower total KL than a model which infers a single latent variable for the whole scene. See the Appendix for a comparison in terms of compression.

In Figure 3, our view traversal (along yaw and position) takes a circular trajectory around the room. Qualitatively, the models show vast differences in their perceived structure of the room. NeRF-VAE blurs out nearly all objects in the scene but understands the geometry of the room and is able to infer wall color. GQN produces more detailed reconstructions, but overfits to particular views and does not interpolate smoothly. SIMONe-VS on the other hand produces sharp reconstructions of fine object structure. Even when it observes objects at a distance or up close, it places and sizes them correctly in totally novel views. This makes SIMONe-VS a powerful choice over NeRF-VAE and GQN-style models when the priority is to capture scene structure across diverse examples.

4.2 Instance segmentation (fully unsupervised)

Having shown the benefit of view information to inferring scene structure in the Section 4.1, we now turn to the added challenge of inferring viewpoint simultaneously with that of scene contents (without supervision).

We demonstrate this by comparing SIMONe to a range of competitive but viewpoint-unaware unsupervised approaches. First, we train two static-frame models: MONet and Slot Attention. MONet uses a similar generative process and training loss to our model, achieving segmentation by

	Static ARI-F			Video ARI-F			
	MONet	SA	S-IODINE	MONet	SA	S-IODINE	SIMONe
Objects Room 9	0.886 (±0.061)	0.784 (±0.138)	0.695 (±0.007)	0.865 (±0.007)	0.066 (±0.014)	0.673 (±0.0.002)	0.936 (±0.010)
CATER	0.937 (±0.004)	0.923 (±0.076)	0.747 (±0.009)	0.412 (±0.012)	0.073 (±0.006)	0.695 (±0.008)	0.918 (±0.036)
Playroom	0.647 (±0.012)	0.653 (±0.024)	0.414 (±0.015)	0.442 (±0.010)	0.059 (±0.002)	0.345 (±0.023)	0.800 (±0.043)

Table 1: **SIMONe segmentation performance** (in terms of Adjusted Rand Index for foreground objects, ARI-F) compared to state of the art unsupervised baselines: two static-frame models (MONet and Slot Attention, SA) and a video model (S-IODINE). We calculate static (S) and video (S) ARI-F scores separately. For static ARI-F, we evaluate the models per still image. For video ARI-F, we evaluate the models across space and time, taking an object's full trajectory as a single class. The video ARI-F thus penalizes models (especially Slot Attention) which fail to track objects stably. We report the mean and standard deviation of scores across 5 random seeds in each case.

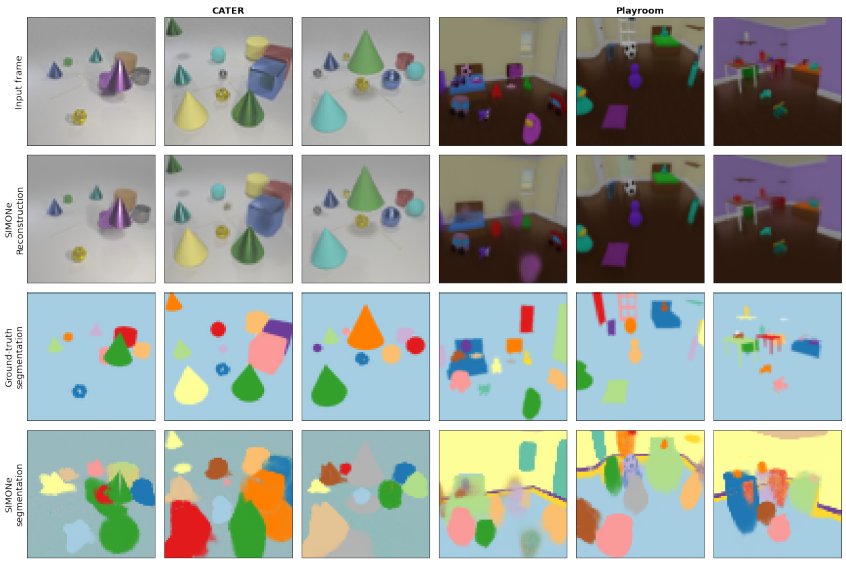

Figure 4: **Segmentations** and reconstructions produced by SIMONe on CATER and Playroom. SIMONe copes well with clutter and different-sized objects. Its masks tend to bound objects loosely.

modeling the scene as a spatial mixture of components, and achieving disentangled representations using a β-weighted KL information bottleneck. On the other hand, it uses a deterministic, recurrent attention network to infer object masks. Slot Attention is a transformer-based autoencoding model which focuses on segmentation performance rather than representation learning. Finally, we also compare against Sequential IODINE ("S-IODINE"), which applies a refinement network to amortize inference over time, separating objects by processing them in parallel. It also uses a β-weighted KL loss to disentangle object representations. Note that S-IODINE is a simplified version of OP3, which additionally attempts to model (pairwise) object dynamics using an agent's actions as inputs. SIMONe and S-IODINE both avoid relying on this privileged information. Table 1 contains a quantitative comparison of segmentation performance across these models, while Figure 4 shows qualitative results.

5 Analysis

5.1 Object and view representations

Qualitative. We visualize the object representations which SIMONe is able to learn across the Playroom's gamut of object types in Figure 5. Note that position and size are well disentangled in each direction. The decoder learns to obey the biases in the dataset–for instance, shelves will slide along a wall when their position latent is traversed. The rubber duck does not morph into a chest

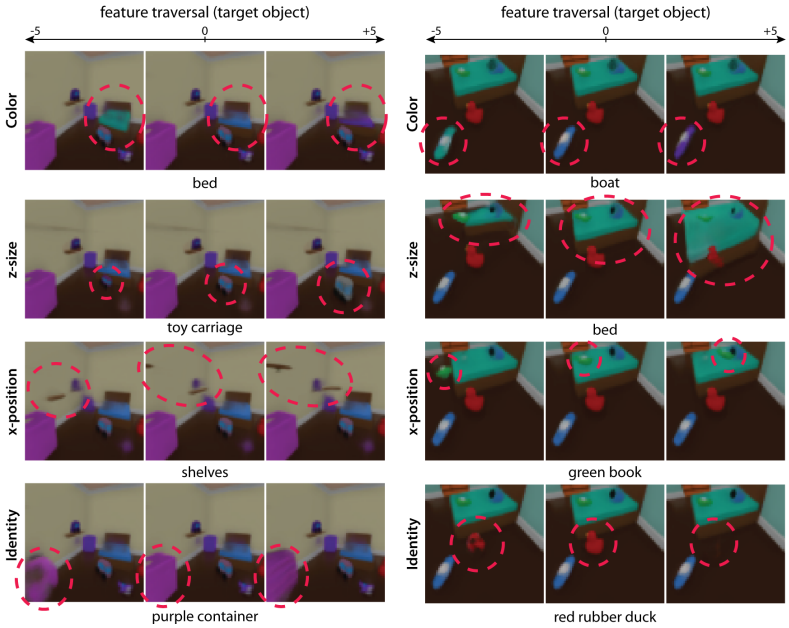

Figure 5: **Object attributes learnt by SIMONe.** In each row, we manipulate a particular object latent for an arbitrary target object in two different scenes, showing the latents' relationship to interpretable attributes like color, position and identity.

	Linear($\mathbf{f}_t$)	MLP($\mathbf{f}_t$)	MLP($\mathbf{o}_1$, ..., $\mathbf{o}_K$)
Camera location	0.832	0.944	-0.017
Camera orientation (Rodrigues)	0.800	0.954	0.272

Table 2: We show that ground-truth camera pose is predictable from the corresponding frame latent, but cannot be predicted from all object latents put together. We report the test R^2 score in each case.

of drawers because those are always located against a wall. This indicates a well-structured latent representation, which the decoder can adapt to.

Quantitative. We assessed frame latents by trying to decode the true camera position and orientation vector from them. We trained linear and MLP regressors to predict the camera pose at time t from the corresponding frame latent f_t on a subset of CATER sequences. We also trained an MLP on the time-invariant object latents $o_{1:K}$ for the same task. We evaluated these decoders on held-out data. Table 2 shows that frame latents describe the viewpoint almost perfectly.

5.2 Temporal abstraction

We observe that SIMONe is capable of summarizing objects' trajectories in their time-invariant latent codes. Figure 6 shows two sequences from the CATER dataset, with object motion trajectories observed from multiple camera trajectories (different than in the original sequences). Note the consistency of relative object positions (at any time-step) from all camera angles. In the single moving object case, its motion could in fact be interpreted as a time-varying global property of the scene. Despite this challenge, SIMONe is able to encode the object's motion as desired in its specific time-invariant code.

6 Discussion and Future Work

Scalability. The transformer-based inference network in SIMONe makes it amenable to processing arbitrarily large videos, just as transformer-based language models can process long text. SIMONe could be trained on windows of consecutive frames sampled from larger videos (aka "chunks"). For inference over a full video, one could add memory slots which carry information over time from one window to the next. Applying SIMONe on sliding windows of frames also presents the opportunity to amortize inference at any given time-step if the windows are partially overlapping

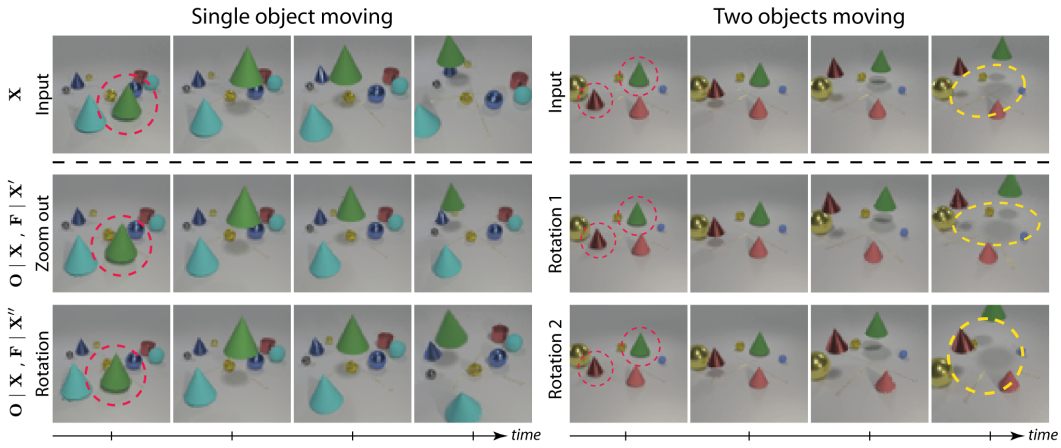

Figure 6: **Separation of object trajectories from camera trajectories.** When encoding a sequence with consistent object dynamics, this information is extracted in the object latents and is unaffected by changing frame latents (see green cone trajectory). On the right panel, the sequencing of object movements is consistently captured, as well as their relative positions (see the pattern of shadows on the floor in the yellow circle).

(so the model could observe every given frame as part of two or more sequences). Our use of the standard transformer architecture also makes SIMONe amenable to performance improvements via alternative implementations.

Limitations. (**1**) SIMONe cannot generate novel videos (e.g. sample a natural camera trajectory via consecutive frame latents) in its current version. This could be addressed in a similar fashion to the way GENESIS [5] built on MONet [3]–it should be possible (e.g. using recurrent networks) to learn conditional priors for objects in a scene and for successive frame latents, which would make SIMONe fully generative. (**2**) We see another possible limitation arising from our strict latent factorization. We have shown that temporally abstracted object features can predict object trajectories when queried by time. This can cover a lot of interesting cases (even multiple object-level "events" over time), but will start to break as object trajectories get more stochastic (i.e. objects transition considerably/chaotically through time). We leave it to future work to explore how temporal abstraction can be combined with explicit per-step dynamics modeling in those cases. For simpler settings, our approach to encoding object trajectories (distilling them across time) is surprisingly effective.

7 Conclusion

We've presented SIMONe, a latent variable model which separates the time-invariant, object-level properties of a video scene from the time-varying, global properties. Our choice of scalable modules such as transformers for inference, and a pixel-wise decoder, allow the model to extract this information effectively.

SIMONe can learn the common structure across a variety of procedurally instantiated scenes. This enables it to recognize and generalize to novel scene instances from a handful of correlated views, as we showcased via 360-degree view interpolation in the view-supervised setting. More significantly, SIMONe can learn to infer the two sets of latent variables jointly without supervision. Aided by cross-frame spatio-temporal attention, it achieves state-of-the-art segmentation performance on complex 3D scenes. Our latent factorization (and information bottleneck pressures) further help with learning meaningful object representations. SIMONe can not only separate static object attributes (like size and position), but it can also separate the dynamics of different objects (as time-invariant localized properties) from global changes in view.

We have discussed how the model can be applied to much longer videos in the future. It also has potential for applications in robotics (e.g. sim-to-real transfer) and reinforcement learning, where view-invariant object information (and summarizing their dynamics) could dramatically improve how agents reason about objects.

References

[1] Jitendra Malik. Scene understanding in the era of deep learning, Jun 2015. URL https://www.robots.ox.ac.uk/seminars/Extra/2015_06_22_JitendraMalik.pdf.

[2] Farzad Husain, Babette Dellen, and Carme Torras. *Chapter 20 - Scene Understanding Using Deep Learning*, pages 373–382. Academic Press, 2017. ISBN 978-0-12-811318-9. doi: https://doi.org/10.1016/B978-0-12-811318-9.00020-X.

[3] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

[4] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International Conference on Machine Learning*, pages 2424–2433. PMLR, 2019.

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

[6] Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. In *International Conference on Learning Representations*, 2020.

[7] S M Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, David P Reichert, Lars Buesing, Theophane Weber, Oriol Vinyals, Dan Rosenbaum, Neil Rabinowitz, Helen King, Chloe Hillier, Matt Botvinick, Daan Wierstra, Koray Kavukcuoglu, and Demis Hassabis. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, June 2018.

[8] Vincent Sitzmann, Michael Zollhoefer, and Gordon Wetzstein. Scene representation networks: Continuous 3D-Structure-Aware neural scene representations. In H Wallach, H Larochelle, A Beygelzimer, F d'Alché Buc, E Fox, and R Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[9] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421. Springer, 2020.

[10] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J.J. Leonard. Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[11] Binbin Xu, Wenbin Li, Dimos Tzoumanikas, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. Mid-fusion: Octree-based object-level multi-instance dynamic slam. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5231–5237. IEEE, 2019.

[12] Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020.

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

[14] Antonia Creswell, Rishabh Kabra, Chris Burgess, and Murray Shanahan. Unsupervised object-based transition models for 3d partially observable environments. *arXiv preprint arXiv:2103.04693*, 2021.

[15] Rishi Veerapaneni, John D Co-Reyes, Michael Chang, Michael Janner, Chelsea Finn, Jiajun Wu, Joshua Tenenbaum, and Sergey Levine. Entity abstraction in visual model-based reinforcement learning. In *Conference on Robot Learning*, pages 1439–1456. PMLR, 2020.

[16] Polina Zablotskaia, Edoardo A Dominici, Leonid Sigal, and Andreas M Lehrmann. Unsupervised video decomposition using spatio-temporal iterative inference. *arXiv preprint arXiv:2006.14727*, 2020.

[17] S M Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E Hinton. Attend, infer, repeat: fast scene understanding with generative models. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 3233–3241, Red Hook, NY, USA, December 2016. Curran Associates Inc.

[18] Adam Kosiorek, Hyunjik Kim, Yee Whye Teh, and Ingmar Posner. Sequential attend, infer, repeat: Generative modelling of moving objects. *Adv. Neural Inf. Process. Syst.*, 31, 2018.

[19] Eric Crawford and Joelle Pineau. Exploiting spatial invariance for scalable unsupervised object tracking. *arXiv preprint arXiv:1911.09033*, 2019.

[20] Eric Crawford and Joelle Pineau. Spatially invariant unsupervised object detection with convolutional neural networks. *AAAI*, 33(01):3412–3420, July 2019.

[21] Jindong Jiang, Sepehr Janghorbani, Gerard de Melo, and Sungjin Ahn. SCALOR: Generative world models with scalable object representations. *arXiv preprint arXiv:1910.02384*, October 2019.

[22] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. *arXiv preprint arXiv:2011.13961*, November 2020.

[23] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. *arXiv preprint arXiv:2011.12948*, 2020.

[24] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for Space-Time view synthesis of dynamic scenes. *arXiv preprint arXiv:2011.13084*, November 2020.

[25] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B Tenenbaum, and Jiajun Wu. Neural radiance flow for 4D view synthesis and video processing. *arXiv preprint arXiv:2012.09790*, December 2020.

[26] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. BARF: Bundle-Adjusting neural radiance fields. *arXiv preprint arXiv:2104.06405*, April 2021.

[27] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, February 2021.

[28] Adam Kosiorek, Heiko Strathmann, Daniel Zoran, Pol Moreno, Rosalia Schneider, Soňa Mokrá, and Danilo Jimenez Rezende. NeRF-VAE: A geometry aware 3D scene generative model. In *International Conference on Machine Learning*. PMLR, 2021.

[29] Alex Trevithick and Bo Yang. GRF: Learning a general radiance field for 3D scene representation and rendering. *arXiv preprint arXiv:2010.04595*, October 2020.

[30] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. *arXiv preprint arXiv:2012.02190*, December 2020.

[31] Michael Niemeyer and Andreas Geiger. GIRAFFE: Representing scenes as compositional generative neural feature fields. *arXiv preprint arXiv:2011.12100*, November 2020.

[32] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. *IEEE Robotics Automation Magazine*, 13(2):99–110, 2006. doi: 10.1109/MRA.2006.1638022.

[33] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (slam): part ii. *IEEE Robotics Automation Magazine*, 13(3):108–117, 2006. doi: 10.1109/MRA.2006.1678144.

[34] Josep Aulinas, Yvan Petillot, Joaquim Salvi, and Xavier Lladó. The slam problem: A survey. In *Proceedings of the 2008 Conference on Artificial Intelligence Research and Development: Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence*, page 363–371, NLD, 2008. IOS Press. ISBN 9781586039257.

[35] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 2016. doi: 10.1177/0278364915620033.

[36] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[37] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[38] Ruihao Li, Sen Wang, and Dongbing Gu. Deepslam: A robust monocular slam system with unsupervised deep learning. *IEEE Transactions on Industrial Electronics*, 68(4):3577–3587, 2021. doi: 10.1109/TIE.2020.2982096.

[39] Mingyang Geng, Suning Shang, Bo Ding, Huaimin Wang, and Pengfei Zhang. Unsupervised learning-based depth estimation-aided visual slam approach. *Circuits, Systems, and Signal Processing*, 39(2): 543–570, 2020.

[40] Diederik P Kingma and Max Welling. An introduction to variational autoencoders. *arXiv preprint arXiv:1906.02691*, 2019.

[41] Adam R Kosiorek, Heiko Strathmann, Daniel Zoran, Pol Moreno, Rosalia Schneider, Soňa Mokrá, and Danilo J Rezende. Nerf-vae: A geometry aware 3d scene generative model. *arXiv preprint arXiv:2104.00587*, 2021.

[42] Rohit Girdhar and Deva Ramanan. CATER: A diagnostic dataset for compositional actions & temporal reasoning. In *International Conference on Learning Representations*, 2020.

[43] Josh Abramson, Arun Ahuja, Arthur Brussee, Federico Carnevale, Mary Cassin, Stephen Clark, Andrew Dudzik, Petko Georgiev, Aurelia Guy, Tim Harley, et al. Imitating interactive intelligence. *arXiv preprint arXiv:2012.05672*, 2020.

[44] Danilo Jimenez Rezende and Fabio Viola. Taming vaes. *arXiv preprint arXiv:1810.00597*, 2018.

[45] Vasu Parameswaran and Rama Chellappa. View invariance for human action recognition. *International Journal of Computer Vision*, 66(1):83–101, 2006.

[46] Yang Liu, Zhaoyang Lu, Jing Li, and Tao Yang. Hierarchically learned view-invariant representations for cross-view action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(8): 2416–2430, 2018.

[47] Ting Liu, Jennifer J Sun, Long Zhao, Jiaping Zhao, Liangzhe Yuan, Yuxiao Wang, Liang-Chieh Chen, Florian Schroff, and Hartwig Adam. View-invariant, occlusion-robust probabilistic embedding for human pose. *arXiv preprint arXiv:2010.13321*, 2020.

[48] Jennifer J Sun, Jiaping Zhao, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, and Ting Liu. View-invariant probabilistic embedding for human pose. In *European Conference on Computer Vision*, pages 53–70. Springer, 2020.

[49] Yana Hasson, Bugra Tekin, Federica Bogo, Ivan Laptev, Marc Pollefeys, and Cordelia Schmid. Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In *CVPR*, 2020.

[50] Emily L Denton and Vighnesh Birodkar. Unsupervised learning of disentangled representations from video. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[51] Yizhe Zhu, Martin Renqiang Min, Asim Kadav, and Hans Peter Graf. S3VAE: Self-supervised sequential VAE for representation disentanglement and data generation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2020.

[52] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. *arXiv preprint arXiv:1802.07687*, February 2018.

[53] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[54] Li Yingzhen and Stephan Mandt. Disentangled sequential autoencoder. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5670–5679. PMLR, 2018.

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[56] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[57] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End object detection with transformers. In *Computer Vision – ECCV 2020*, pages 213–229. Springer International Publishing, 2020.

[58] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. MaX-DeepLab: End-to-End panoptic segmentation with mask transformers. *arXiv preprint arXiv:2012.00759*, December 2020.

[59] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H S Torr, and Li Zhang. Rethinking semantic segmentation from a Sequence-to-Sequence perspective with transformers. *arXiv preprint arXiv:2012.15840*, December 2020.

[60] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

[61] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is Space-Time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, February 2021.

[62] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *arXiv preprint arXiv:2102.00719*, February 2021.

[63] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A video vision transformer. *arXiv preprint arXiv:2103.15691*, March 2021.

[64] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-End video instance segmentation with transformers. *arXiv preprint arXiv:2011.14503*, November 2020.

[65] Nicholas Watters, Loic Matthey, Christopher P Burgess, and Alexander Lerchner. Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes. *arXiv preprint arXiv:1901.07017*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] If we've overestimated the scope at all, it was in an honest attempt to connect to prior work. We were careful not to overclaim and careful not to re-invent the wheel.
   (b) Did you describe the limitations of your work? [Yes] See Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix A.6.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] All training instructions, hyperparameters, and evaluation schemes are detailed in the Appendix. The code will be released after reviews.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix A.3 and A.4.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Table 1 and Appendix.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix A.3 and A.4.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4.
(b) Did you mention the license of the assets? [Yes] See Appendix A.2.
(c) Did you include any new assets either in the supplemental material or as a URL? [No]
(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]