SlotFormer: Unsupervised Visual Dynamics Simulation with Object-Centric Models

Ziyi Wu[1,2], Nikita Dvornik[3,1], Klaus Greff[4], Thomas Kipf*[4], Animesh Garg*[1,2]

[1] University of Toronto [2] Vector Institute
[3] Samsung AI Centre Toronto [4] Google Research

Abstract. Understanding dynamics from visual observations is a challenging problem that requires disentangling individual objects from the scene and learning their interactions. While recent object-centric models can successfully decompose a scene into objects, modeling their dynamics effectively still remains a challenge. We address this problem by introducing SlotFormer – a Transformer-based autoregressive model operating on learned object-centric representations. Given a video clip, our approach reasons over object features to model spatio-temporal relationships and predicts accurate future object states. In this paper, we show that the unsupervised SlotFormer's dynamics model can be used to improve the performance on supervised downstream Visual Question Answering (VQA) tasks. It enables VQA models to reason about the future without object-level labels, even outperforming counterparts that use ground-truth annotations. [1]

1 Introduction

The ability to understand complex systems and interactions between its elements is a key component of intelligent systems. Learning the dynamics of a multi-object systems from visual observations entails capturing *object* instances, their appearance, position and motion, and simulating their spatio-temporal interactions. One approach to visual dynamics modeling is to frame it as a prediction problem directly in the pixel space [15,20,3]. This paradigm builds on global frame-level representations, and uses dense feature maps of past frames to predict future features. By design, such models are object-agnostic, treating background and foreground modeling as equal. This frequently results in poorly learned object dynamics, producing unrealistic future predictions over longer horizons [12]. Another perspective to dynamics learning is through object-centric dynamics models [8,17,9]. This class of methods first represents a scene as a set of object-centric features (a.k.a. slots), and then learns the interactions among the slots to model scene dynamics. It allows for more natural dynamics modeling and leads to more faithful simulation [19,23]. To achieve this goal, earlier object-centric models bake in strong scene [5] or object [10] priors in their frameworks, while more recent methods [7,23] learn object interactions purely from

* Equal advisory contribution

[1] Additional results and details are available at our Website. For experimental results of SlotFormer on more downstream tasks, please refer to our full paper [21].

data, with the aid of Graph Neural Networks (GNNs) [1] or Transformers [18]. Yet, these approaches independently model the per-frame object interactions and their temporal evolution, using different networks. This suggests that a simpler and more effective dynamics model is yet to be designed.

In this work, we argue that learning a system's dynamics from video effectively requires two key components: i) *strong unsupervised object-centric representations* (to capture objects in each frame) and ii) a *powerful dynamical module* (to simulate spatio-temporal interactions between the objects). To this end, we propose SlotFormer: a simple and effective Transformer-based object-centric dynamics model, which builds upon object-centric features [6,16], and requires no human supervision. We treat dynamics modeling as a sequential learning problem: given a sequence of input images, SlotFormer takes in the object-centric representations extracted from these frames, and predicts the object features in the future steps. By conditioning on multiple frames, our method is capable of capturing the spatio-temporal object relationships simultaneously, thus ensuring consistency of object properties and motion in the synthesized frames. In summary, this work makes the following contributions:

1. SlotFormer: a Transformer-based model for object-centric visual simulation;
2. Our method achieves state-of-the-art results on CLEVRER VQA task, without leveraging any object-level annotations. This proves that SlotFormer's unsupervised dynamics knowledge can be successfully transferred to downstream supervised tasks to improve their performance "for free".

2 SlotFormer: Object-Oriented Dynamics Learning

In this section, we describe our Transformer-based autoregressive model for dynamics learning. Taking T video frames as inputs, SlotFormer first leverages a pre-trained object-centric model to extract object features (a.k.a. slots) from each frame (Section 2.1). These slots are then forwarded to the Transformer module for joint spatio-temporal reasoning, and used to predict future slots (Section 2.2). The whole pipeline is trained by minimizing reconstruction loss in both feature and image space (Section 2.3). We show the overall model architecture in Figure 1.

2.1 Slot-based Object-Centric Representation

We build on the Slot Attention architecture to extract slots from videos due to their strong performance in unsupervised object discovery. Given T input frames $\{\boldsymbol{x}_t\}_{t=1}^{T}$, our object-centric model first extracts image features using a Convolutional Neural Network (CNN) encoder, then adds positional encodings, and flattens them into a set of vectors $\boldsymbol{h}_t \in \mathbb{R}^{M \times D_{enc}}$, where M is the size of the flattened feature grid and D_{enc} is the feature dimension. Then, the model initializes N slots $\tilde{\mathcal{S}}_t \in \mathbb{R}^{N \times D_{slot}}$ from a set of learnable vectors ($t = 1$), and performs Slot Attention [11] to update the slot representations as $\mathcal{S}_t = f_{SA}(\tilde{\mathcal{S}}_t, \boldsymbol{h}_t)$. Here, f_{SA} binds slots to objects via iterative Scaled Dot-Product Attention [18], encouraging scene decomposition. To achieve temporal alignment of slots, $\tilde{\mathcal{S}}_t$ for $t \geq 2$ is initialized as $\tilde{\mathcal{S}}_t = f_{trans}(\mathcal{S}_{t-1})$, where f_{trans} is the transition function implemented as a Transformer encoder.

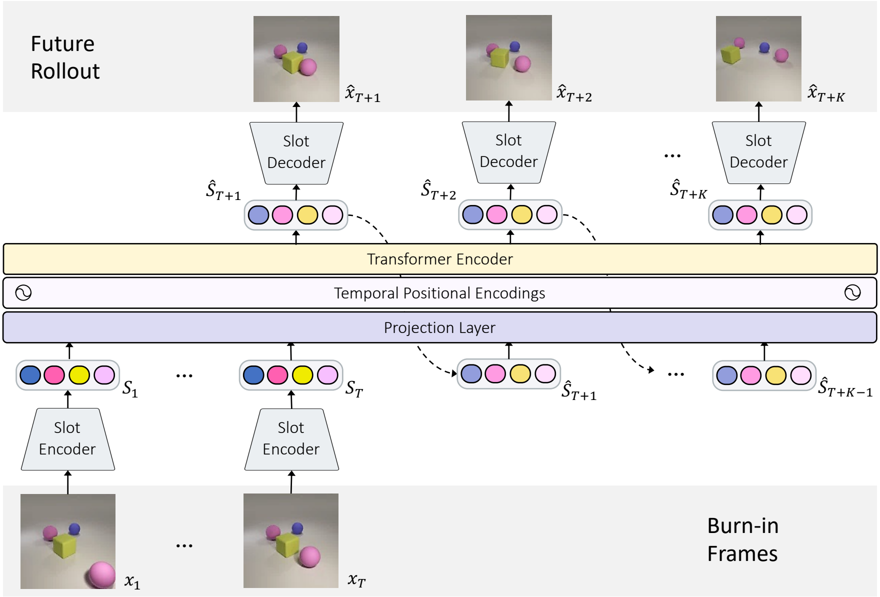

Fig. 1: SlotFormer architecture overview. Taking multiple video frames $\{\boldsymbol{x}_t\}_{t=1}^T$ as input, we first extract object slots $\{\mathcal{S}_t\}_{t=1}^T$ using the pretrained object-centric model. Then, slots are linearly projected and added with temporal positional encoding. The resulting tokens are fed to the Transformer module to generate future slots $\{\hat{\mathcal{S}}_{T+k}\}_{k=1}^K$ in an autoregressive manner.

Before training the Transformer-based dynamics model, we first pre-train the object-centric model using reconstruction loss on videos from the target dataset. This ensures the learned slots can accurately capture both foreground objects and background environment of the scene.

2.2 Dynamics Prediction with Autoregressive Transformer

Overview. Given slots $\{\mathcal{S}_t\}_{t=1}^T$ extracted from T video frames, SlotFormer is able to synthesize a sequence of future slots $\{\mathcal{S}_{T+k}\}_{k=1}^K$ for any given horizon K. Our model operates by alternating between two steps: i) feed the slots into a Transformer that performs joint spatio-temporal reasoning and predicts slots at the next timestep, $\hat{\mathcal{S}}_{t+1}$, ii) feed the predicted slots back into the Transformer to keep generating future rollout autoregressively. See Figure 1 for the pipeline overview of our method.

Architecture. To build the SlotFormer's dynamics module, $\mathcal{T}$, we adopt the standard Transformer encoder module with $N_{\mathcal{T}}$ layers. To match the inner dimensionality D_e of $\mathcal{T}$, we linearly project the input sequence of slots to a latent space $G_t = \text{Linear}(\mathcal{S}_t) \in \mathbb{R}^{N \times D_e}$. To indicate the order of input slots, we add positional encoding (P.E.) to the latent embeddings. A naive solution would be to add a sinusoidal positional encoding to every slot regardless of its timestep, as done in [4]. However, this would break the *permutation equivariance* among slots, which is a useful property of our model. Therefore, we only apply positional encoding at the temporal level, such that the slots at the same timestep receives the same positional encoding:

$$V = [G_1, G_2, ..., G_T] + [P_1, P_2, ..., P_T], \tag{1}$$

where $V \in \mathbb{R}^{(TN)\times D_e}$ is the resulting input to the transformer $\mathcal{T}$ and $P_t \in \mathbb{R}^{N\times D_e}$ denotes the sinusoidal positional encoding duplicated N times. As we will show in the ablation study, the temporal positional encoding enables better prediction results despite having fewer parameters.

Now, we can utilize the Transformer $\mathcal{T}$ to reason about the dynamics of the scene. Denote the Transformer output features as $U = [U_1, U_2, ..., U_T] \in \mathbb{R}^{(TN)\times D_e}$, we take the last N features $U_T \in \mathbb{R}^{N\times D_e}$ and feed them to a linear layer to obtain the predicted slots at timestep $T+1$:

$$U = \mathcal{T}(V), \quad \hat{\mathcal{S}}_{T+1} = \text{Linear}(U_T). \tag{2}$$

For consequent future predictions, $\hat{\mathcal{S}}_{T+1}$ will be treated as the ground-truth slots along with $\{\mathcal{S}_t\}_{t=2}^{T}$ to predict $\hat{\mathcal{S}}_{T+2}$. In this way, the Transformer can be applied autoregressively to generate any given number, K, of future frames, as illustrated in Figure 1.

Remark. The SlotFormer's architecture allows to *preserve temporal consistency among slots* at different timesteps. To realize such consistency, we employ residual connections from $\mathcal{S}_t$ to $\hat{\mathcal{S}}_{t+1}$, which forces the Transformer $\mathcal{T}$ to apply refinement to the slots while preserving their absolute order. Owing to this order invariance, SlotFormer can be used to reason about individual object's dynamics for long-term rollout, and can be seamlessly integrated with downstream task models.

2.3 Model Training

In contrast to prior research that predicts image tokens one by one with a causal attention mask in GPT-style, we generate all the slots at the next timestep in parallel. Therefore, we do not need the teacher forcing strategy [14] for training. Instead, we train the model using the predicted slots as inputs. This simulates the error accumulation process in long-term sequence generation, and improves the quality of the generated videos, as we will show in our experiments.

For training, we use a slot reconstruction loss (in L_2) denoted as:

$$\mathcal{L}_S = \frac{1}{K \cdot N} \sum_{k=1}^{K} \sum_{n=1}^{N} ||\hat{\boldsymbol{s}}_{T+k}^{n} - \boldsymbol{s}_{T+k}^{n}||^2. \tag{3}$$

When using SAVi as the object-centric model, we also employ an image reconstruction loss to promote prediction of consistent object attributes such as colors and shapes. The predicted slots are decoded to images by the frozen SAVi decoder f_{dec}, and then matched to the original frames as:

$$\mathcal{L}_I = \frac{1}{K} \sum_{k=1}^{K} ||f_{dec}(\hat{\mathcal{S}}_{T+k}) - \boldsymbol{x}_{T+k}||^2. \tag{4}$$

The final objective function is a weighted combination of the two losses with a hyper-parameter λ:

$$\mathcal{L} = \mathcal{L}_S + \lambda \mathcal{L}_I. \tag{5}$$

Method	Descriptive	Explanatory		Predictive		Counterfactual		Average
		per opt.	per ques.	per opt.	per ques.	per opt.	per ques.	
Aloe	95.04	98.08	94.88	93.11	87.28	90.82	74.09	87.82
Aloe + **Ours**	95.17	98.04	94.79	**96.50**	**93.29**	90.63	73.78	**89.26**

Table 1: Accuracy of different questions on CLEVRER. All numbers are in %.

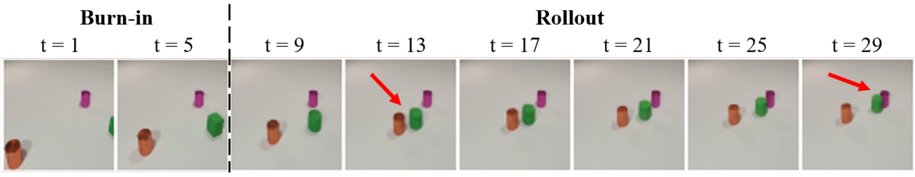

Fig. 2: Qualitative results on CLEVRER VQA task. To answer the question "*Will the green object collide with the purple cylinder?*", SlotFormer successfully simulates the first collision between the green and the brown cylinder (t = 13), which leads to the second collision between the target objects (t = 29).

3 Experiments

Dataset. We demonstrate SlotFormer's ability for downstream reasoning task on the *CLEVRER* [22] VQA dataset. CLEVRER provides four types of questions: descriptive, explanatory, predictive and counterfactual. The predictive questions require the model to simulate future interactions of objects such as collisions. Therefore, we focus on the accuracy improvement on predictive questions by using SlotFormer's future rollout.

Implementation details. We first pre-train SAVi [6] on the videos to perform unsupervised segmentation, and then extract slots for training SlotFormer. We use $N = 7$ slots and slot size $D_{slot} = 128$, and train SAVi on video clips of 6 consecutive frames. Other training settings follow the original paper. We use $T = 6$ burn-in steps and $K = 10$ rollout steps to train SlotFormer. Following previous work [23], we subsample the video by a factor of 2. For the Transformer, we set the latent size $D_e = 256$, and stack $N_{\mathcal{T}} = 4$ layers. We train SlotFormer with a batch size of 128 using the Adam optimizer for 500k steps. The learning rate first linearly warms up to $2e-4$, and then decays to zero in a cosine manner.

Task model. Since SlotFormer is a generic dynamics model, we can combine it with any reasoning module to enhance its performance. We choose *Aloe* [4] as it can jointly process slots and texts. To answer the predictive questions, we explicitly unroll SlotFormer for 32 steps, and run Aloe on the predicted future slots. For other questions, we simply apply Aloe on slots from the observed frames. We re-implement Aloe in PyTorch [13] with the same hyper-parameters and training settings to get a similar performance. Since our SAVi slot representations are more powerful than their MO-Net [2] slots, we only need 12 layers of Transformer encoder, while they use 28 layers.

Results. Table 1 presents the results for the Aloe baseline and Aloe incorporated with SlotFormer. We focus our comparison on the predictive question accuracy. The dynamics predicted by SlotFormer improves the accuracy of Aloe by 3.4% and 6.0% in the per option (per opt.) and per question (per ques.) setting, respectively. On the CLEVRER public leaderboard predictive question subset, we rank first in the per option setting, and second in the per question setting. As a fully unsupervised dynamics model, our method even outperforms previous state-of-the-art DCL and VRDP which use supervisedly trained object detectors. Figure 2 shows an example of our predicted dynamics, where SlotFormer accurately simulates two consecutive collision events.

4 Conclusion

In this paper, we propose SlotFormer, a Transformer-based autoregressive model that enables consistent long-term dynamics modeling with object-centric representations. SlotFormer learns complex spatio-temporal interactions between the objects and generates accurate future states. Moreover, SlotFormer can transfer unsupervised dynamics knowledge to downstream (supervised) reasoning tasks which leads to state-of-the-art results on CLEVRER VQA task. Finally, we believe that unsupervised object-centric dynamics models hold great potential for simulating complex datasets, advancing world models, and reasoning about the future with minimal supervision; and that SlotFormer is a new step towards this goal.

References

1. Battaglia, P.W., Hamrick, J.B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al.: Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261 (2018)
2. Burgess, C.P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M., Lerchner, A.: Monet: Unsupervised scene decomposition and representation. arXiv preprint arXiv:1901.11390 (2019)
3. Denton, E., Fergus, R.: Stochastic video generation with a learned prior. In: ICML. pp. 1174–1183. PMLR (2018)
4. Ding, D., Hill, F., Santoro, A., Reynolds, M., Botvinick, M.: Attention over learned object embeddings enables complex visual reasoning. NeurIPS **34** (2021)
5. Jiang, J., Janghorbani, S., De Melo, G., Ahn, S.: Scalor: Generative world models with scalable object representations. In: ICLR (2019)
6. Kipf, T., Elsayed, G.F., Mahendran, A., Stone, A., Sabour, S., Heigold, G., Jonschkowski, R., Dosovitskiy, A., Greff, K.: Conditional object-centric learning from video. arXiv preprint arXiv:2111.12594 (2021)
7. Kipf, T., van der Pol, E., Welling, M.: Contrastive learning of structured world models. arXiv preprint arXiv:1911.12247 (2019)
8. Kosiorek, A., Kim, H., Teh, Y.W., Posner, I.: Sequential attend, infer, repeat: Generative modelling of moving objects. NeurIPS **31** (2018)
9. Kossen, J., Stelzner, K., Hussing, M., Voelcker, C., Kersting, K.: Structured object-aware physics prediction for video modeling and planning. In: ICLR (2019)
10. Lin, Z., Wu, Y.F., Peri, S., Fu, B., Jiang, J., Ahn, S.: Improving generative imagination in object-centric world models. In: ICML. pp. 6140–6149. PMLR (2020)
11. Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., Kipf, T.: Object-centric learning with slot attention. NeurIPS **33**, 11525–11538 (2020)

12. Oprea, S., Martinez-Gonzalez, P., Garcia-Garcia, A., Castro-Vargas, J.A., Orts-Escolano, S., Garcia-Rodriguez, J., Argyros, A.: A review on deep learning techniques for video prediction. TPAMI (2020)
13. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. NeurIPS **32** (2019)
14. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018)
15. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. NeurIPS **28** (2015)
16. Singh, G., Wu, Y.F., Ahn, S.: Simple unsupervised object-centric learning for complex and naturalistic videos. arXiv preprint arXiv:2205.14065 (2022)
17. van Steenkiste, S., Chang, M., Greff, K., Schmidhuber, J.: Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. In: ICLR (2018)
18. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. NeurIPS **30** (2017)
19. Veerapaneni, R., Co-Reyes, J.D., Chang, M., Janner, M., Finn, C., Wu, J., Tenenbaum, J., Levine, S.: Entity abstraction in visual model-based reinforcement learning. In: CoRL. pp. 1439–1456. PMLR (2020)
20. Wang, Y., Long, M., Wang, J., Gao, Z., Yu, P.S.: Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. NeurIPS **30** (2017)
21. Wu, Z., Dvornik, N., Greff, K., Kipf, T., Garg, A.: Slotformer: Unsupervised visual dynamics simulation with object-centric models. arXiv preprint arXiv:2210.05861 (2022)
22. Yi, K., Gan, C., Li, Y., Kohli, P., Wu, J., Torralba, A., Tenenbaum, J.B.: Clevrer: Collision events for video representation and reasoning. In: ICLR (2019)
23. Zoran, D., Kabra, R., Lerchner, A., Rezende, D.J.: Parts: Unsupervised segmentation with slots, attention and independence maximization. In: ICCV. pp. 10439–10447 (2021)