Reasoning-Enhanced Object-Centric Learning for Videos

Jian Li
lijian2022@ruc.edu.cn
Renmin University of China
Beijing, China

Pu Ren
ren.pu@northeastern.edu
Northeastern University
Boston, MA, USA

Yang Liu
liuyang22@ucas.ac.cn
University of Chinese Academy of Sciences
Beijing, China

Hao Sun*
haosun@ruc.edu.cn
Renmin University of China
Beijing, China

Abstract

Object-centric learning aims to break down complex visual scenes into more manageable object representations, enhancing the understanding and reasoning abilities of machine learning systems toward the physical world. Recently, slot-based video models have demonstrated remarkable proficiency in segmenting and tracking objects, but they overlook the importance of the effective reasoning module. In the real world, reasoning and predictive abilities play a crucial role in human perception and object tracking; in particular, these abilities are closely related to human intuitive physics. Inspired by this, we designed a novel reasoning module called the Slot-based Time-Space Transformer with Memory buffer (STATM) to enhance the model's perception ability in complex scenes. The memory buffer primarily serves as storage for slot information from upstream modules, the Slot-based Time-Space Transformer makes predictions through slot-based spatiotemporal attention computations and fusion. Our experimental results on various datasets indicate that the STATM module can significantly enhance the capabilities of multiple state-of-the-art object-centric learning models for video. Moreover, as a predictive model, the STATM module also performs well in downstream prediction and Visual Question Answering (VQA) tasks. We will release our codes and data at *https://github.com/intell-sci-comput/STATM*.

CCS Concepts

• **Computing methodologies** → **Computer vision**; *Artificial intelligence*; *Machine learning*.

Keywords

Object-Centric Learning, Slot-based Spatiotemporal Attention, Intuitive Physics, Spatiotemporal Prediction

---

*Corresponding author.

1 Introduction

Objects are the fundamental elements that constitute our world, which adhere to the fundamental laws of physics. Humans learn through activities such as observing the world and interacting with it. They utilize the knowledge acquired via these processes for reasoning and prediction. All these aspects are crucial components of human intuitive physics [30, 32, 44, 46]. Therefore, object-centric research is pivotal for comprehending human cognitive processes and for developing more intelligent artificial intelligence (AI) systems. By studying the properties, movements, interactions, and behaviors of objects, we can uncover the ways and patterns in which humans think and make decisions in the domains of perception, learning, decision-making, and planning. This contributes to the advancement of more sophisticated machine learning algorithms and AI systems, enabling them to better understand and emulate human intuitive physical abilities [21, 49].

Recently, the representative SAVi [27] and SAVi++ [15] models have demonstrated impressive performance in object perception. SAVi (Slot Attention for Video) employed optical flow as a prediction target and leveraged a small set of abstract hints as conditional inputs in the first frame to acquire object-centric representations of dynamic scenes. SAVi++ (Towards End-to-End Object-Centric Learning from Real-World Videos) enhanced the SAVi by integrating depth prediction and implementing optimal strategies for architectural design and data augmentation. Both SAVi and SAVi++ execute two steps on observed video frames: a prediction step and a correction step. The correction step uses inputs to update the slots. The prediction step uses the slots information of the objects provided by the correction step for prediction. The predictor's output initializes the correction process in the subsequent time step, ensuring the model's consistent ability to track objects over time.

The two main steps of such a model operate in a positive feedback loop. The more accurate the predictions, the better the corrections become. Consequently, the more accurate the corrections, the more precise the information provided for the prediction step is, leading to better predictions. Therefore, having a reasonable and efficient predictor is crucial for the entire model. In real-world scenarios,
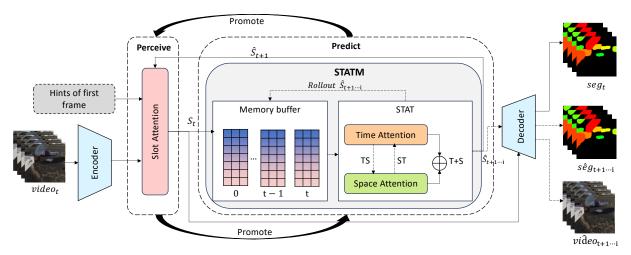
Figure 1: Slot-based Time-Space Transformer with Memory buffer architecture overview. The model employs Slot Attention [36] for perception, which utilizes slot information predicted by STATM predictor from previous timestep and features extracted by encoder to update slot information. For the first frame, the initial slot information is obtained through either Gaussian distribution or hints module. The updated slot information is then stored in a memory buffer for subsequent use by the TATM. TATM performs reasoning by incorporating temporal cross-attention and spatial self-attention. The integration of temporal and spatial attention can be achieved in various ways. STATM supports both single-step predictions and long-sequence rollouts, where single-step prediction results can be used by Slot Attention to update slot information, and long-sequence rollout results can be used to downstream tasks such as VQA. Both perceptual and predicted slot information can be used by the decoder to obtain reconstruction results and segmentation masks. The architecture features perception and prediction modules that mutually enhance each other.

humans also engage in prediction as a crucial aspect of their object perception and tracking, but their prediction behaviors often involve more intricate processes. Humans typically combine the motion state of an object with the interactions of other objects to predict possible future states and positions of the object. The object's motion state is inferred by humans using their common sense from the object's past positions over a while. In so doing, humans enhance their ability to recognize and track relevant objects within complex scenes, which is an integral component of human intuitive physics [40, 48, 50]. The prediction step in SAVi and SAVi++ is similar to human inference, but their predictor module is somewhat simplistic, as it relies solely on single-frame information from the current time step for prediction.

In the field of object-centric video prediction, SlotFormer [60] and OCVP [53] transform the spatiotemporal attention structure used for video classification in TimeSformer [3] into a similar structure utilized for slot prediction. However, the number of slots (also considered as tokens) required by SlotFormer is likely to increase dramatically with time as the number of objects in the scene grows, leading to an excess of superfluous slot computation. OCVP explored two types of spatiotemporal slot models, which, to some extent, mitigated the increase in token numbers, yet still faced the issue of unnecessary slot computations. The predictive capability of both approaches heavily depends on the quality of the upstream slot extraction. Neither approach made improvements to the upstream module responsible for slot extraction, nor did they delve into the impact of prediction on upstream perception.

Human perception and prediction are typically complementary. When assessing an object's movement, humans often rely on short-term memory impressions of the object. These impressions, along with consideration of environmental factors within the scene, are used to predict the object's movement. Through a comprehensive analysis of time and space, humans anticipate the object's next position. Drawing inspiration from human behavior, we introduce a novel prediction module aimed at enhancing slot-based models for video. This module comprises two key components: 1) **Slot-based Memory Buffer**: designed to store historical slot information obtained from the upstream module. 2) **Slot-based Time-Space Transformer Module**: designed by applying spatiotemporal attention mechanisms to slots for inferring the temporal motion states of objects and calculating spatial objects interactions, which also integrates time and space attention results. The module only computes using slots from the current moment and those from past moments. This not only addresses the issue of the increasing number of tokens as time progresses and the number of objects in the scene increases, but it also reduces unnecessary slot computations. We term the proposed model as *Slot-based Time-Space Transformer with Memory buffer* (STATM). Upon substituting the prediction module of SAVi and SAVi++ into the STATM, we observe distinct impacts of different spatiotemporal fusion methods on SAVi and SAVi++. By employing an appropriate fusion method and memory buffer sizes, we observed a significant enhancement in the object segmentation and tracking capabilities of SAVi and SAVi++ on videos containing complex backgrounds and multiple objects per scene.

Overall, our contributions are summarized as follows:

- We have investigated the impact of prediction modules on models utilizing Slot Attention [36] for object-centric learning from video and have developed the STATM module as a predictor. By simply incorporating a memory buffer and spatiotemporal attention, we have significantly enhanced the capabilities of models like SAVi [27], and SAVi++[15].
- We have diligently worked to reduce the computational cost of the spatiotemporal module. In contrast to other models that utilize multiple frames for prediction [53, 60], our spatiotemporal module only combines current and past slots for computation, effectively decreasing the number of tokens required for the prediction module and the amount of slot computations it performs. As time progresses and the number of objects in the scene increases, the advantage of STATM becomes even more apparent.
- We have conducted experiments across multiple benchmarks. We have observed that the STATM module significantly enhances the capabilities of multiple state-of-the-art object-centric learning models for video, such as SAVi [27], SAVi++ [15], and STEVE [45]. Moreover, as a predictive model, the STATM also performs well in downstream prediction and Visual Question Answering (VQA) tasks.
- Furthermore, we have briefly explored the impact of various spatiotemporal architectures and different memory buffer sizes on model performance.

2 Related Work

Object-centric Learning. In recent years, object-centric learning has emerged as a significant research direction in computer vision and machine learning. It aims to enable machines to perceive and understand the environment from an object-centered perspective, thereby constructing more intelligent visual systems. There is a rich literature on this research, including SQAIR [29], R-SQAIR [47], SCALOR [22], Monet [4], OP3 [52], ViMON [58], PSGNet [2], SIMONe [24], and others [25, 28, 61, 68]. Slot-based Models represent a prominent approach within object-centric learning. They achieve this by representing each object in a scene as an individual slot, which is used to store object features and attributes [13, 17, 31, 60, 63].

Slot-based Attention and spatiotemporal Attention. Our current work is closely related to slot-based attention and spatiotemporal attention. There are a lot of works related to slot-based attention [18, 36, 54, 60, 65, 71]. Spatiotemporal attention mechanisms are particularly effective in handling video data or time-series data, allowing networks to understand and leverage relationships between different time steps or spatial positions [3, 33, 38]. Currently, they find wide applications in various fields such as video object detection and tracking [6, 34], action recognition [64], natural language processing [59, 62], medical image processing [69], among many others [9, 12, 67].

Prediction and Inference on Physics. The implementation of object-centric physical reasoning is crucial for human intelligence and is also a key objective in artificial intelligence. Interaction Network [1] as the first general-purpose learnable physics engine, is capable of performing reasoning tasks centered around objects

or relationships. Another similar study is the Neural Physics Engine [5]. On the other hand, Visual Interaction Networks [57] can learn physical laws from videos to predict the future states of objects. Additionally, there are many models developed based on this foundation [7, 14, 23, 39, 41, 45]. Additionally, there are many object-centric predictive models that are based on slot representations [10, 11, 60]. However, their performance largely depends on the quality of slot extraction by upstream perception modules. In order to achieve a deeper understanding of commonsense intuitive physics within artificial intelligence systems, [41] have built a system capable of learning various physical concepts, albeit requiring access to privileged information such as segmentation. Our research primarily aims to construct an object-centric system for object perception, learning of physics, and reasoning.

3 Slot-based Time-Space Transformer with Memory Buffer (STATM)

To enhance the slot-based video models, we introduce a new module called *Slot-based Time-Space Transformer with Memory Buffer* (STATM) as the predictor. This module consists of two key components: 1) memory buffer, and 2) Slot-based Time-Space Transformer (STAT). The memory buffer serves as a repository for storing historical slot information obtained from upstream modules, while STAT utilizes the information stored in the memory buffer for prediction and causal reasoning. The overall framework is shown in Figure 1.

3.1 Memory Buffer

The memory module is utilized for storing slot information from the upstream modules. We employ a queue-based storage mechanism. The representation of the memory buffer at time t is given by:

$$M_t = Queue(S_i, \ldots, S_t), \tag{1}$$

where $S_t = \{s_{(0,t)}, \ldots, s_{(N,t)}\}$ represents the slot information extracted from the corrector module at time t. Here, N signifies the number of slots, which is associated with the number of objects within the scene. The size of M can be fixed or infinite. The new information is appended at the end of the queue.

3.2 Slot-based Time-Space Transformer

The primary role of STAT (Slot-based Time-Space Transformer) lies in leveraging slot data from the memory buffer to facilitate slot-based dynamic temporal reasoning and spatial interaction computations. Furthermore, it integrates the outcomes of temporal reasoning and spatial interactions to achieve a unified understanding. Specifically, for temporal dynamic reasoning, a cross-attention mechanism is employed, which effectively utilizes historical context stored in the memory buffer to enable accurate predictions of future states. Meanwhile, for spatial interaction computations, we employ a self-attention mechanism that operates on slot representations to compute the relevance between different slots within the S. The results obtained from temporal dynamic reasoning and spatial interaction computation are merged to provide a holistic understanding encompassing both temporal dynamics and spatial interactions. This comprehensive representation enhances the model's capability for accurate prediction and reasoning in object-centric tasks.

(a) Corresponding Slot Attention (CS)
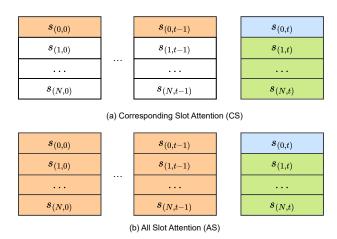

(b) All Slot Attention (AS)

Figure 2: Spatiotemporal attention computation architectures. The green slots represent those employed for spatial attention computation, while the orange slots are indicative of those used for temporal attention computation.

We propose three approaches:

$$\widehat{S}_{t+1} = CrossAtt^{time}(S_t, M_t) + SelfAtt^{space}(S_t) \tag{2a}$$

$$\widehat{S}_{t+1} = CrossAtt^{time}(SelfAtt^{space}(S_t), M_t) \tag{2b}$$

$$\widehat{S}_{t+1} = SelfAtt^{space}(CrossAtt^{time}(S_t, M_t)), \tag{2c}$$

(2a) $T+S$: The sum of computed temporal attention and spatial attention. (2b) ST: Spatial attention computation followed by using the outcome as input for temporal attention. (2c) TS: Temporal attention computation followed by using the outcome as input for spatial attention.

As shown in Figure 2, we introduce two computational architectures for spatiotemporal attention: (a) *Corresponding Slot Attention (CS)*: For slot $s_{(i,t)}$, temporal attention is computed by using it and corresponding slots in $\{s_{(i,0)}, \ldots, s_{(i,t-1)}\}$, while spatial attention computation is performed using it and all slots within $\{s_{(0,t)}, \ldots, s_{(N,t)}\}$. (b) *All Slot Attention (AS)*: For slot $s_{(i,t)}$, temporal attention is computed by using it and all slots in $\{s_{(0,0)}, \ldots, s_{(N,t-1)}\}$. The spatial attention computation remains the same as in the CS.

In the CS architecture, $s_{(i,t)}$ undergoes temporal attention computation exclusively with its corresponding slots. This design offers several notable advantages. Firstly, it enables a more robust association between objects and slots in terms of temporal sequences, preserving the slot's invariance with respect to the object. Additionally, this approach significantly reduces computational costs when compared to the AS structure. This efficiency makes the CS architecture an appealing choice for achieving effective temporal binding while optimizing computational resources.

In the AS architecture, the temporal attention involves calculating the attention between $s_{(i,t)}$ and all previous slots.

3.3 Differences between STATM and Other Slot-based Models

In the field of object-centric video prediction, SlotFormer [60] and OCVP [53] transform the spatiotemporal attention structure used

for video classification in TimeSformer [3] into a similar structure utilized for slot prediction. Nevertheless, the computational slots utilized by these models are likely to surge dramatically over time as the number of objects in the scene increases, leading to an excess of unnecessary slot computations.

Our model design is focused on reducing the computational cost. Traditionally, slot-based models, such as SlotFormer[60], flatten the time T ($T > 2$) and the number of slots N ($N > 2$) when using T time steps for prediction, then input them into the transformer encoder to calculate full attention between each slot. Thus, the transformer encoder must process $T \times N$ tokens and perform attention calculations on $T \times N$ slots, leading to $(T \times N)^2$ calculations.

In contrast, our spatiotemporal modules only combine the current time slots with the previous time slots for calculation. Therefore, the prediction modules based on the CS structure only requires $T + N$ tokens, and the AS structure requires $T \times N$ tokens. In attention calculations, we only consider computations between the current moment slots and other slots. For the CS structure, in terms of time attention, it only needs to calculate attention between current time slot s_i ($i = 1, \ldots, N$) and the other $T - 1$ slots, and in terms of space attention, it only needs to calculate attention between N slots, thus involving $N \times (T - 1) + N^2$ slot calculations in total. For the AS structure, it involves $(T - 1) \times N \times N$ calculations in time and N^2 calculations in space, totaling $T \times N^2$ calculations. Clearly, aside from the token requirement for the AS structure, both the number of tokens and the amount of computation required by our spatiotemporal attention module are significantly less than that of SlotFormer. We also found that using only the CS structure can achieve good model performance. As the number of frames and objects in a scene increase, the CS structure is more efficient.

3.4 Training

Our focus is on conducting perception and prediction experiments. In the perception experiments, we intend to enhance several state-of-the-art object-centric learning models using the STATM module. Consequently, the training loss for each model may differ. We train STATM-SAVi and STATM-SAVi++ to minimize the L2 loss between the predicted and ground-truth targets, such as optical flow, images, and depth signals. The training loss for STATM-STEVE aligns with the loss function used in STEVE[45].

In the prediction experiments, we train the STATM model by jointly minimizing slots and images reconstruction loss (also L2).

4 Experiments

Our experiments primarily comprise three parts: perception, prediction and VQA, and ablation. The perception experiments aim to verify the impact of the STATM module on existing state-of-the-art object-centric learning models for video. The prediction experiments are designed to preliminarily demonstrate the robust performance of the STATM module as a predictive model in downstream prediction and VQA tasks. The ablation studies focus on assessing the effects of the memory buffer and various spatiotemporal structures on model performance.

Baselines. In the perception experiments, we primarily compare SAVi [27], SAVi++ [15], STEVE [45] and SAVi-SlotFormer. For SAVi, we chose official implementation, SAVi-small, as the baseline. The

SAVi-samll model includes five components: encoder, decoder, slot initialization, corrector, and predictor. The encoder uses a CNN to extract features from video frames. Slot initialization, using either an MLP or a CNN, prepares slots with initial data like bounding boxes. The corrector, powered by Slot Attention [36], updates slots using encoder features. The predictor, a transformer block, uses self-attention for forecasting and initializes the corrector for consistent tracking. Finally, the decoder outputs RGB predictions and an alpha mask using a Spatial Broadcast Decoder. The SAVi-SlotFormer is a baseline we develop to assess the impact of the prediction module on the perceptual performance of SAVi. SAVi++ [15] has a structure similar to SAVi with a ResNet34 backbone. Unlike SAVi, SAVi++ introduces depth as self-supervised objectives. It also incorporates data augmentation and utilizes a transformer encoder after ResNet34.

In the prediction section, we compare G-SWM [35], SAVi-dyn[60], and SlotFormer [60]. SlotFormer [60] is a transformer-based framework for object-centric visual simulation. It leverages slots extracted by upstream modules like SAVi to train a slot-based transformer encoder model for prediction purposes.

For the VQA experiments, our main comparisons involve DCL [8], VRDP [11], and SlotFormer [60]. SlotFormer utilizes prediction results from rollout simulations to train Aloe [10] for VQA tasks. VRDP [11] is designed to jointly learn visual concepts and infer physics models of objects and their interactions from both videos and language. It primarily consists of three modules: a visual perception module, a concept learner, and a differentiable physics engine. We have implemented VRDP with a visual perception module that is trained based on object properties.

For more baselines, please refer to Appendix Section A.

Metrics. To evaluate the model's object-centric learning capability for video, we selected the Adjusted Rand Index (ARI) [19, 42] and the mean Intersection over Union (mIoU) as evaluation metrics. ARI quantifies the alignment between predicted and ground-truth segmentation masks. For scene decomposition assessment, we commonly employ FG-ARI and, which is a permutation-invariant clustering similarity metric. It allows us to compare inferred segmentation masks to ground-truth masks while excluding background pixels. mIoU is a widely used segmentation metric that calculates the mean Intersection over Union values for different classes or objects in a segmentation task. To assess video quality, we report PSNR, SSIM [55], and LPIPS [70]. To evaluate the prediction outcomes, we utilize FG-ARI and FG-mIoU.

Datasets. To evaluate the object-centric learning capability, we utilized the synthetic Multi-Object Video (MOVi) datasets [16, 43]. These datasets are divided into five distinct categories: A, B, C, D, and E. MOVi-A and B depict relatively straightforward scenes, each containing a maximum of 10 objects. MOVi-C, D, and E present more intricate scenarios with complex natural backgrounds. MOVi-C, generated using a stationary camera, presents scenes with up to 10 objects. Transitioning to MOVi-D, the dataset extends the object count to accommodate a maximum of 23 objects. Lastly, MOVi-E introduces an additional layer of complexity by incorporating random linear camera movements. Each video sequence is sampled at a frame rate of 12, resulting in a total of 24 frames per second.

To assess the predictive and Visual Question Answering (VQA) capabilities, we have selected the CLEVRER [66] dataset. CLEVRER dataset is specifically designed for video understanding and reasoning, focusing on the dynamics of objects and their causal interactions. For the VQA task, CLEVRER incorporates four types of questions: descriptive (e.g., "what color"), explanatory ("what's responsible for"), predictive ("what will happen next"), and counterfactual ("what if"). The predictive questions require the model to simulate future object interactions, such as collisions. Thus, we are particularly concentrating on enhancing the accuracy of predictive questions through the implementation of STATM's future rollout.

Training Setup. In all experiments except the ablation study in Section 4.3, we used the STAT encoding block in combination with the CS attention architecture, featuring the T+S spatiotemporal fusion approach. For perception experiments, we utilized videos comprising of 6 frames at a resolution of 64×64 pixels to train the STATM-SAVi and SAVi models. The training process is conducted over 100k iterations. Similarly, the STATM-SAVi++ and SAVi++ models were trained on continuous videos consisting of 6 frames at a higher resolution of 128×128 pixels, with training duration encompassing 100k or 500k iterations. The buffer size was unconstrained during training, and the maximum length of effective information was limited to 6 due to the utilization of a 6-frame training sequence. Bounding boxes were used as the conditioning for all models. For prediction and VQA experiments, we train our models (STATM-SAVi) for 400k steps with a batch size of 64 on the CLEVRER dataset to extract slots. The number of slots is set to 7, with a learning rate of 0.0001. We subsample the video by a factor of 2 to train STATM, conducting approximately 500k training steps with a batch size of 64 and a learning rate of 0.0002. We use rollout slots to train Aloe [10], targeting around 300k steps with a learning rate of 0.0001 and a batch size of 128. We use the Adam optimizer and apply warm-up and decay learning rate schedule for the first 2.5% of the total training steps. For more training setup, please refer to Appendix Section A and B.

4.1 Perception

Results. Quantitative results can be seen in Table 1 and 2, and qualitative results in Figure 3. From Table 1 and 2, it is evident that STATM can significantly enhance the performance of existing state-of-the-art models. Comparing SAVi, SAVi-SlotFormer and STATM-SAVi, SAVi performs reasonably well on simple datasets but struggles with complex datasets. When SlotFormer is used as the predictor, it enhances SAVi's performance on complex datasets, yet it reduces its effectiveness on simpler datasets. Conversely, when STATM serves as the predictor, it not only improves SAVi's performance on complex datasets but also maintains its performance on simple datasets. In comparison between SAVi++ and STATM-SAVi++, the enhanced model shows a notable improvement, and except for the MOVi-E (due to insufficient training), it can generally match or surpass the optimal results of SAVi++. Further, when considering the best results of STATM-SAVi++* against the official results SAVi++, our model's performance is markedly superior to the original model, indicating that the model's benefits do not diminish with increased training. The limitations of depth information and overfitting mentioned in the SAVi++ [15] do not appear in STATM-SAVi++. The qualitative results from Figure 3 also demonstrate the superior performance of our models (e.g., the notable

Table 1: Enhancement results by STATM on models with hints. The first five rows depict the evaluation results for models trained for 100k steps with a batch size of 32. SAVi-SlotFormer* denotes our implemented baseline model. SAVi++* represents results from SAVi++ paper [15]. STATM-SAVi++* denotes the evaluation results for STATM-SAVi++ model trained for 500k steps with a batch size of 64 (Mean ± standard error over 3 seeds).

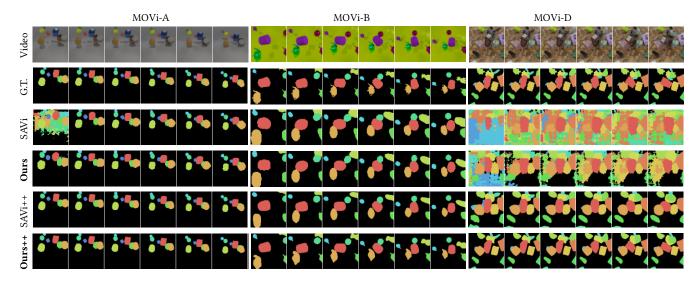| Model	mIoU↑ (%)					FG-ARI↑ (%)				
	A	B	C	D	E	A	B	C	D	E
SAVi	62.8	41.6	22.0	6.8	4.0	**91.1**	70.2	50.4	18.4	10.8
SAVi-SlotFormer*	63.5	-	-	-	7.5	86.4	-	-	-	31.2
STATM-SAVi (Ours)	**67.5**	**42.8**	**34.0**	**17.0**	**9.0**	**91.1**	**70.7**	**57.7**	**40.9**	**36.9**
SAVi++	82.8	52.5	47.8	43.6	26.1	96.7	78.5	76.3	81.5	81.7
STATM-SAVi++ (Ours)	**83.5**	**52.5**	**49.5**	**50.1**	**27.9**	**96.9**	**78.9**	**77.7**	**85.8**	**85.0**
SAVi++*	76.1±0.9	25.8±11.3	45.2±0.1	48.3±0.5	47.1±1.3	98.2±0.2	48.3±15.7	81.9±0.2	86.0±0.3	84.1±0.9
STATM-SAVi++* (Ours)	**85.6±0.6**	**60.4±1.2**	**52.4±0.2**	**57.0±0.4**	**55.4±0.9**	**98.3±0.2**	**84.9±2.5**	**82.2±0.2**	**89.1±0.2**	**88.6±0.5**

Figure 3: Qualitative results of our model compared to SAVi and SAVi++ on the MOVi dataset. Compared with SAVi and SAVi++, our model is slightly better than the SAVi/SAVi++ mode on the relatively simple datasets. As the complexity of the datasets increases, the advantage of our model becomes more pronounced.

Table 2: Enhancement Results (FG-ARI%) on STEVE.

Model	MOVi-D	MOVi-E
STEVE	47.67	52.15
STATM-STEVE (Ours)	**51.73**	**55.78**

difference between SAVi and STATM-SAVi in the first frame of MOVi-A).

In addition, we also tried to use the STATM module to improve the unsupervised scene segmentation model STEVE. The results can be found in Table 2. We found that STAM remains effective for unsupervised object-centric learning model.

From Figure 4, it is evident that both SAVi and SAVi++ face challenges in recognizing newly appearing objects and objects that reappear after being occluded. When a new object emerges, the original algorithm often misidentifies it as background or an already

occupied slot is taken over by the new object. With the incorporation of STATM, although the model may not immediately segment the new object, as the historical information accumulates in the memory buffer, the STAT module can gradually provide the corrector with hints required for the object segmentation (e.g., position or shape), eventually, the model can successfully segment the object. When an object disappears (possibly temporarily occluded), SAVi++ immediately releases the slot associated with the object, potentially causing difficulty in binding the object to the original slot or even failing to recognize it upon reappearance. For STATM-SAVi++, due to the presence of the memory buffer and temporal attention in STATM, when the occluded object reappears, it can easily be assigned to its historical slot. Only when the object has been absent for an extended period will the slot be released.

Memory Cost and Inference Time. The memory cost and inference time (taken to process 250 videos) are listed in Table 3. We can see that adding STATM to the model doesn't bring much

(a) New object appear   (b) Object reappears after being occluded
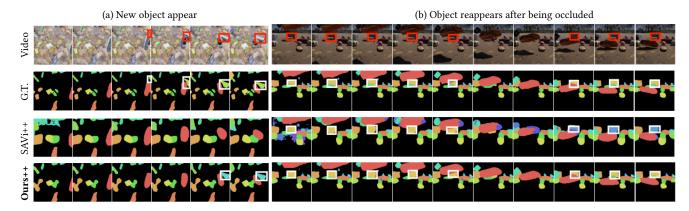

Figure 4: Qualitative results of our model compared to SAVi++. (a) When a new object appears, the SAVi++ cannot recognize it, but our model can correctly identifies it after 1-2 frames. (b) When an object reappears after being obscured, the SAVi++ either assigns it to a different slot (color change) or fails to recognize it. In contrast, our model can correctly identify it.

Table 3: Memory cost and inference time on MOVi-A and E validation sets, with each set containing 250 videos, each video contains 24 frames (batch size of 32, on one A100 GPU).

Model	Memory (GB)		Infer. Time (s)	
	A	E	A	E
SAVi	25.91	56.42	78.4	141.3
SAVi-SlotFormer	25.92	56.43	102.6	204.1
STATM-SAVi (Ours)	25.91	56.42	81.4	146.4

Table 4: Generalization results on MOVi datasets.

Model	mIoU↑ (%)			FG-ARI↑ (%)		
	C	D	E	C	D	E
SAVi-S (IID)	22.0	6.8	4.0	50.4	18.4	10.8
SAVi-S (OOD)	21.1	6.3	3.7	52.8	19.8	9.9
STATM-SAVi-S (IID)	34.0	17.0	9.0	57.7	40.9	36.9
STATM-SAVi-S (OOD)	33.2	17.0	8.2	59.7	43.4	35.9
SAVi++ (IID)	47.8	43.6	26.1	76.3	81.5	81.7
SAVi++ (OOD)	46.9	44.0	25.5	77.7	82.2	82.5
STATM-SAVi++ (IID)	49.5	50.1	27.9	77.7	85.8	85.0
STATM-SAVi++ (OOD)	48.7	50.3	27.4	78.9	86.4	85.8

extra memory cost, inference time and parameters. Detailed comparison of parameters can be found in Appendix Table S1 Appendix Section C.

Generalization. We selected the models trained with a batch size of 32 and 100k training steps to assess its generalization. The test sets utilized the default test split of MOVi-C, D and E dataset, featuring scenes exclusively consist of held-out objects and background images to evaluate generalization. The results are presented in Table 4 and Figure 6 (a).

Discussion. Certainly, the STATM module, acting as a predictor, significantly enhances the model performance of SAVi and SAVi++. Compared to the original models, the improved model can achieve good performance with fewer training steps and a smaller batch

Table 5: Perception results on CLEVRER.

Model	MSE↓	FG-ARI(%)	FG-mIoU(%)
SAVi	0.24	91.4	**77.6**
STATM-SAVi (Ours)	**0.15**	**93.5**	**77.6**

size. STATM-SAVi++ also addresses the overfitting issue of SAVi++ on simple datasets (especially noticeable in MOVi-B). The enhanced models also exhibit good generalization. Importantly, the integration of a STAT encoding block does not lead to a significant increase in memory cost and inference time.

4.2 Prediction and VQA

In this section, our primary objective is to succinctly validate the performance of STATM in downstream prediction and VQA tasks.

Perception on CLEVRER. Due to the effectiveness of object-centric predictive models largely depends on the quality of slots extracted by upstream perceptual modules. Additionally, MOVi validation sets only includes sequences of 24 frames in length. Therefore, in Table 5, we demonstrate a comparison of perceptual effects on CLEVER with longer sequences of 128 frames. It is evident that STATM-SAVi achieves a significantly higher FG-ARI compared to SAVi, indicating that STATM provides a more pronounced enhancement on SAVi in longer sequences on relatively simple datasets.

Prediction. Table 6 displays the evaluation results of visual quality and object dynamics on CLEVRER. It is evident that when using slots extracted by SAVi to train our STATM for prediction, the model shows improvement in all metrics except PSNR, indicating a certain advantage of our model in prediction tasks. When using slots extracted by STATM-SAVi to train STATM for predictions, both video quality and predictive metrics see further enhancements, particularly FG-ARI. This further demonstrates the advantages of our model in both perception and prediction.

Figure 5 displays generation results of long-sequence prediction on CLEVRER. We observe that SlotFormer performs well in shorter sequence predictions. However, as time progresses, SlotFormer's performance deteriorates significantly, exhibiting blurriness, incorrect dynamics, and inaccurate colors, even becoming completely
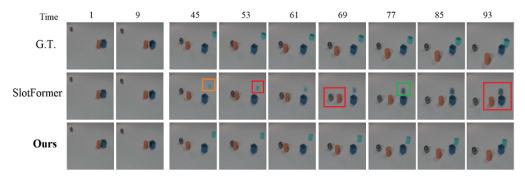
Figure 5: Results of long-sequence prediction on CLEVRER. After surpassing a certain time point, results generated by SlotFormer in prediction clearly begin to deviate from the ground truth, exhibiting artifacts such as blurry (orange boxes), incorrect dynamics (red boxes), and inaccurate colors (green boxes). Meanwhile, our model demonstrates good performance.

Table 6: Evaluations of visual quality (columns 2-4) and object dynamics (columns 5-6) on CLEVRER. SA+ST and ST+ST respectively represent the results of using SAVi and STATM-SAVi for slot extraction followed by training STATM.

Model	PSNR	SSIM	LPIPS↓	FG-ARI(%)	FG-mIoU(%)
SAVi-Dyn	29.77	**0.89**	0.19	64.32	18.25
SlotFormer	30.21	**0.89**	**0.11**	63.00	49.40
SA+ST(Ours)	30.10	**0.89**	**0.11**	63.11	49.55
ST+ST(Ours)	**30.22**	**0.89**	**0.11**	**64.56**	**49.57**

Table 7: Predictive VQA accuracy on CLEVRER.

Model	per opt.(%)	per ques.(%)
DCL	90.52	82.03
VRDP	95.68	91.35
SlotFormer	96.50	93.29
STATM (Ours)	**96.62**	**93.63**

inconsistent with the ground truth. In contrast, our model continues to perform well.

VQA. In Table 7, we present the accuracy on predictive questions. Notably, also as an unsupervised predictive model, our method surpasses the previous state-of-the-art SlotFormer [60]. Furthermore, on the publicly available CLEVRER leaderboard for the predictive question subset, our approach achieved first rank in the per option setting and second rank in the per question setting.

Discussion. Clearly, our model demonstrates distinct advantages in perception over longer sequences, as well as in downstream prediction and VQA tasks.

4.3 Ablation Study

In this section, we aim to evaluate the influence of different components of STATM, using STATM-SAVi as a baseline.

Memory Buffer. We have designed two sets of experiments to evaluate the impact of the memory buffer: 1) In the first set, we allowed an unlimited memory buffer length during training, but restricted it to a fixed length during testing. 2) In the second set, we fixed the buffer length during training, and removed any buffer length restrictions during testing. To facilitate evaluation, we have

not only assessed the model trained with 6 frames but also extended the training frames to 12. We show the results in Figure 6 (b) and (c).
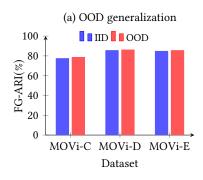
Training phase. From Figure 6 (c), we observe that during the perception training phase, changes in buffer size have minimal impact on the model's performance on simple datasets. For complex datasets, initially increasing the buffer size improves model performance, but further increases eventually lead to a decline. Thus, during the perception training phase, buffer size can be set to 6.
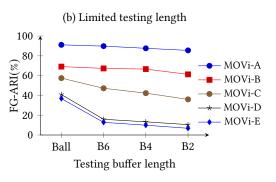
Testing phase. Since the MOVi validation set contains only 24 frames, it does not adequately demonstrate the impact of memory size during the testing phase. Therefore, in Table 8 perception column, we present evaluation results on CLEVRER dataset with a sequence of 128 frames during testing phase. We observe that increasing the buffer size during the testing phase initially improves perceptual outcomes, but beyond a certain size, it has negligible impact. Thus, we set the buffer size to 24 during the testing phase to avoid unnecessary computational costs.

Table 8 prediction and VQA column display the effects of using different buffer sizes of STATM-SAVi to extract slots during the testing phase on prediction and VQA performance. It is observed that slots extracted by SAVi-STATM using different buffer sizes have almost no impact on STATM's predictive capabilities. The impact of buffer size on VQA is minor, but beyond a certain threshold, VQA performance actually deteriorates. Therefore, for prediction and VQA tasks, we use 24 buffer size of STATM-SAVi to extract slot.

In summary, the buffer size can be set to 6 during the training of both the perception model and the long-term reasoning component. When testing the perception model, the buffer size should be set to 24. For the long-term reasoning component during use, a buffer size of 6 yields good results.

Spatiotemporal Fusion and Computation. In Table 9, we display results for STATM-SAVi using different spatiotemporal attention computation and fusion methods. From the first three rows of the table, it can be seen that the T+S spatiotemporal fusion method is relatively superior to both ST and TS. The last two rows indicate that the CS spatiotemporal attention structure is superior to AS. This may be due to the AS structure causing slot confusion from excessive slot interactions. Therefore, we use STAT with CS and T+S for our experiments by default.
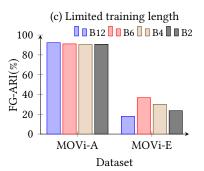
Figure 6: (a) Results on out-of-distribution evaluation splits with new objects and backgrounds. (b) Ablation study of the buffer size on model performance during testing phase. The x-axis represents buffer sizes during testing. (c) Ablation study of the impact of buffer size on model performance during training phase. The different bars represent buffer sizes.

Table 8: Ablation study of buffer on CLEVRER during the testing phase. 'Perception' represents perceptual results of STATM-SAVi with different buffer sizes. 'Prediction' shows prediction results of STATM trained with slots extracted by STATM-SAVi with different buffer sizes. 'VQA' indicates our model's accuracy on predictive questions when using various buffer sizes.

Model	Perception		Prediction		VQA	
	FG-ARI(%)	FG-mIoU(%)	FG-ARI(%)	FG-mIoU(%)	per opt.(%)	per ques.(%)
STATM (24)	93.5	77.6	64.6	49.6	96.62	93.63
STATM (32)	93.5	77.6	64.5	49.5	96.41	93.09
STATM (48)	93.5	77.6	64.6	49.6	96.07	92.67
STATM (128)	93.5	77.6	64.2	49.3	96.21	92.90

Table 9: Ablation study of perception for different spatiotemporal attention computation and fusion methods.

Model	mIoU↑ (%)		FG-ARI↑ (%)	
	A	E	A	E
STATM-SAVi (CS, ST)	58.4	-	90.9	-
STATM-SAVi (CS, TS)	61.2	-	89.7	-
STATM-SAVi (CS, T+S)	67.5	8.5	91.1	36.8
STATM-SAVi (AS, T+S)	-	3.8	-	12.2

4.4 Limitations

We did not assess our model using real-world datasets. Our perception and prediction models are trained separately, although they share a common predictive structure. In the future, we are more interested in implementing integrated training of our models.

5 Conclusion

In the real world, all objects follow the laws of physics. Intuitive physics serves as the bridge and connection through which humans comprehend the world. Our research aims to construct an object-centric system for object perception, learning of physics, and reasoning to explore whether deep learning models can learn physical concepts like humans, and use these learned physical laws to make inferences and predictions about the future motion of objects. We have designed a more reasonable prediction module called STATM, which clearly improved slot-based models in the context of scene understanding and prediction. We demonstrated that reasoning and prediction abilities influence each other. Through a series

of experiments, we have shown the advantages of our model in tasks such as perception, prediction, and VQA. We also explore the impact of different spatiotemporal attention and fusion methods, and memory buffer on model perception and prediction. Although many challenges still remain in this field, the results presented in this paper illustrate that well-designed deep learning models can mimic human perception and prediction. In the future, we hope to implement joint training of our perception and prediction models, along with real-time perception and prediction, and validate the model in real-world scenarios.

Acknowledgments

References

[1] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. 2016. Interaction networks for learning about objects, relations and physics. *Advances in neural information processing systems* 29 (2016).
[2] Daniel Bear, Chaofei Fan, Damian Mrowca, Yunzhu Li, Seth Alter, Aran Nayebi, Jeremy Schwartz, Li F Fei-Fei, Jiajun Wu, Josh Tenenbaum, et al. 2020. Learning physical graph representations from visual scenes. *Advances in Neural Information Processing Systems* 33 (2020), 6027–6039.
[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. 2021. Is space-time attention all you need for video understanding?. In *ICML*, Vol. 2. 4.

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

[5] Michael B Chang, Tomer Ullman, Antonio Torralba, and Joshua B Tenenbaum. 2016. A compositional object-based approach to learning physical dynamics. *arXiv preprint arXiv:1612.00341* (2016).

[6] Beijing Chen, Tianmu Li, and Weiping Ding. 2022. Detecting deepfake videos based on spatiotemporal attention and convolutional LSTM. *Information Sciences* 601 (2022), 58–70.

[7] Chang Chen, Fei Deng, and Sungjin Ahn. 2021. Roots: Object-centric representation and rendering of 3d scenes. *The Journal of Machine Learning Research* 22, 1 (2021), 11770–11805.

[8] Zhenfang Chen, Jiayuan Mao, Jiajun Wu, Kwan-Yee Kenneth Wong, Joshua B Tenenbaum, and Chuang Gan. 2021. Grounding physical concepts of objects and events through dynamic visual reasoning. *arXiv preprint arXiv:2103.16564* (2021).

[9] Dawei Cheng, Sheng Xiang, Chencheng Shang, Yiyi Zhang, Fangzhou Yang, and Liqing Zhang. 2020. Spatio-temporal attention-based neural network for credit card fraud detection. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 362–369.

[10] David Ding, Felix Hill, Adam Santoro, Malcolm Reynolds, and Matt Botvinick. 2021. Attention over learned object embeddings enables complex visual reasoning. *Advances in neural information processing systems* 34 (2021), 9112–9124.

[11] Mingyu Ding, Zhenfang Chen, Tao Du, Ping Luo, Josh Tenenbaum, and Chuang Gan. 2021. Dynamic visual reasoning by learning differentiable physics models from video and language. *Advances In Neural Information Processing Systems* 34 (2021), 887–899.

[12] Yukai Ding, Yuelong Zhu, Jun Feng, Pengcheng Zhang, and Zirun Cheng. 2020. Interpretable spatio-temporal attention LSTM model for flood forecasting. *Neurocomputing* 403 (2020), 348–359.

[13] Andrea Dittadi, Samuele Papa, Michele De Vita, Bernhard Schölkopf, Ole Winther, and Francesco Locatello. 2021. Generalization and robustness implications in object-centric learning. *arXiv preprint arXiv:2107.00637* (2021).

[14] Danny Driess, Zhiao Huang, Yunzhu Li, Russ Tedrake, and Marc Toussaint. 2023. Learning multi-object dynamics with compositional neural radiance fields. In *Conference on Robot Learning*. PMLR, 1755–1768.

[15] Gamaleldin Elsayed, Aravindh Mahendran, Sjoerd van Steenkiste, Klaus Greff, Michael C Mozer, and Thomas Kipf. 2022. Savi++: Towards end-to-end object-centric learning from real-world videos. *Advances in Neural Information Processing Systems* 35 (2022), 28940–28954.

[16] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. 2022. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3749–3761.

[17] Mohammed Hassanin, Saeed Anwar, Ibrahim Radwan, Fahad S Khan, and Ajmal Mian. 2022. Visual attention methods in deep learning: An in-depth survey. *arXiv preprint arXiv:2204.07756* (2022).

[18] Jiaying Hu, Yan Yang, Chencai Chen, Liang He, and Zhou Yu. 2020. SAS: Dialogue state tracking via slot attention and slot information sharing. In *Proceedings of the 58th annual meeting of the association for computational linguistics*. 6366–6375.

[19] Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification* 2 (1985), 193–218.

[20] Daniel Im, Sungjin Ahn, Roland Memisevic, and Yoshua Bengio. 2017. Denoising criterion for variational auto-encoding framework. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 31.

[21] Michael Janner, Sergey Levine, William T. Freeman, Joshua B. Tenenbaum, Chelsea Finn, and Jiajun Wu. 2019. Reasoning About Physical Interactions with Object-Oriented Prediction and Planning. In *International Conference on Learning Representations*.

[22] Jindong Jiang, Sepehr Janghorbani, Gerard De Melo, and Sungjin Ahn. 2019. Scalor: Generative world models with scalable object representations. *arXiv preprint arXiv:1910.02384* (2019).

[23] Marko Jusup, Petter Holme, Kiyoshi Kanazawa, Misako Takayasu, Ivan Romić, Zhen Wang, Sunčana Geček, Tomislav Lipić, Boris Podobnik, Lin Wang, et al. 2022. Social physics. *Physics Reports* 948 (2022), 1–148.

[24] Rishabh Kabra, Daniel Zoran, Goker Erdogan, Loic Matthey, Antonia Creswell, Matt Botvinick, Alexander Lerchner, and Chris Burgess. 2021. Simone: View-invariant, temporally-abstracted object representations via unsupervised video decomposition. *Advances in Neural Information Processing Systems* 34 (2021), 20146–20159.

[25] Daniel Kahneman, Anne Treisman, and Brian J Gibbs. 1992. The reviewing of object files: Object-specific integration of information. *Cognitive psychology* 24, 2 (1992), 175–219.

[26] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[27] Thomas Kipf, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. 2021. Conditional Object-Centric Learning from Video. In *International Conference on Learning Representations*.

[28] Thomas Kipf, Elise Van der Pol, and Max Welling. 2019. Contrastive learning of structured world models. *arXiv preprint arXiv:1911.12247* (2019).

[29] Adam Kosiorek, Hyunjik Kim, Yee Whye Teh, and Ingmar Posner. 2018. Sequential attend, infer, repeat: Generative modelling of moving objects. *Advances in Neural Information Processing Systems* 31 (2018).

[30] James R Kubricht, Keith J Holyoak, and Hongjing Lu. 2017. Intuitive physics: Current research and controversies. *Trends in cognitive sciences* 21, 10 (2017), 749–759.

[31] Adarsh Kumar, Peter Ku, Anuj Goyal, Angeliki Metallinou, and Dilek Hakkani-Tur. 2020. Ma-dst: Multi-attention-based scalable dialog state tracking. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 8107–8114.

[32] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. Building machines that learn and think like people. *Behavioral and brain sciences* 40 (2017), e253.

[33] Jun Li, Xianglong Liu, Wenxuan Zhang, Mingyuan Zhang, Jingkuan Song, and Nicu Sebe. 2020. Spatio-temporal attention networks for action recognition and detection. *IEEE Transactions on Multimedia* 22, 11 (2020), 2990–3001.

[34] Lei Lin, Weizi Li, Huikun Bi, and Lingqiao Qin. 2021. Vehicle trajectory prediction using LSTMs with spatial–temporal attention mechanisms. *IEEE Intelligent Transportation Systems Magazine* 14, 2 (2021), 197–208.

[35] Zhixuan Lin, Yi-Fu Wu, Skand Peri, Bofeng Fu, Jindong Jiang, and Sungjin Ahn. 2020. Improving generative imagination in object-centric world models. In *International conference on machine learning*. PMLR, 6140–6149.

[36] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. 2020. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems* 33 (2020), 11525–11538.

[37] Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983* (2016).

[38] Yingtao Luo, Qiang Liu, and Zhaocheng Liu. 2021. Stan: Spatio-temporal attention network for next location recommendation. In *Proceedings of the web conference 2021*. 2177–2185.

[39] Chuizheng Meng, Sungyong Seo, Defu Cao, Sam Griesemer, and Yan Liu. 2022. When physics meets machine learning: A survey of physics-informed machine learning. *arXiv preprint arXiv:2203.16797* (2022).

[40] Alex Mitko and Jason Fischer. 2020. When it all falls down: the relationship between intuitive physics and spatial cognition. *Cognitive research: principles and implications* 5 (2020), 1–13.

[41] Luis S Piloto, Ari Weinstein, Peter Battaglia, and Matthew Botvinick. 2022. Intuitive physics learning in a deep-learning model inspired by developmental psychology. *Nature human behaviour* 6, 9 (2022), 1257–1267.

[42] William M Rand. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association* 66, 336 (1971), 846–850.

[43] Google Research. 2020. *Google Scanned Objects*. https://app.ignitionrobotics.org/GoogleResearch/fuel/collections/Google%20Scanned%20Objects

[44] Ronan Riochet, Mario Ynocente Castro, Mathieu Bernard, Adam Lerer, Rob Fergus, Véronique Izard, and Emmanuel Dupoux. 2018. Intphys: A framework and benchmark for visual intuitive physics reasoning. *arXiv preprint arXiv:1803.07616* (2018).

[45] Gautam Singh, Yi-Fu Wu, and Sungjin Ahn. 2022. Simple unsupervised object-centric learning for complex and naturalistic videos. *Advances in Neural Information Processing Systems* 35 (2022), 18181–18196.

[46] Brian Cantwell Smith. 2019. *The promise of artificial intelligence: reckoning and judgment*. Mit Press.

[47] Aleksandar Stanić and Jürgen Schmidhuber. 2019. R-sqair: Relational sequential attend, infer, repeat. *arXiv preprint arXiv:1910.05231* (2019).

[48] Erik Blaine Sudderth. 2006. *Graphical models for visual object recognition and tracking*. Ph. D. Dissertation. Massachusetts Institute of Technology.

[49] Qu Tang, Xiangyu Zhu, Zhen Lei, and Zhaoxiang Zhang. 2023. Intrinsic Physical Concepts Discovery with Object-Centric Predictive Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 23252–23261.

[50] Tomer D Ullman, Elizabeth Spelke, Peter Battaglia, and Joshua B Tenenbaum. 2017. Mind games: Game engines as an architecture for intuitive physics. *Trends in cognitive sciences* 21, 9 (2017), 649–665.

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

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

[53] Angel Villar-Corrales, Ismail Wahdan, and Sven Behnke. 2023. Object-Centric Video Prediction Via Decoupling of Object Dynamics and Interactions. In *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE, 570–574.

[54] Yanbo Wang, Letao Liu, and Justin Dauwels. 2023. Slot-VAE: Object-Centric Scene Generation with Slot Attention. *arXiv preprint arXiv:2306.06997* (2023).

[55] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions*

on image processing 13, 4 (2004), 600–612.

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

[57] Nicholas Watters, Daniel Zoran, Theophane Weber, Peter Battaglia, Razvan Pascanu, and Andrea Tacchetti. 2017. Visual interaction networks: Learning a physics simulator from video. *Advances in neural information processing systems* 30 (2017).

[58] Marissa A Weis, Kashyap Chitta, Yash Sharma, Wieland Brendel, Matthias Bethge, Andreas Geiger, and Alexander S Ecker. 2020. Unmasking the inductive biases of unsupervised object representations for video sequences. *arXiv preprint arXiv:2006.07034* 2 (2020).

[59] Henry Weld, Xiaoqi Huang, Siqu Long, Josiah Poon, and Soyeon Caren Han. 2022. A survey of joint intent detection and slot filling models in natural language understanding. *Comput. Surveys* 55, 8 (2022), 1–38.

[60] Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. 2023. SlotFormer: Unsupervised Visual Dynamics Simulation with Object-Centric Models. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=TFbwV6I0VLg

[61] Junyu Xie, Weidi Xie, and Andrew Zisserman. 2022. Segmenting moving objects via an object-centric layered representation. *Advances in Neural Information Processing Systems* 35 (2022), 28023–28036.

[62] Weijia Xu, Batool Haider, and Saab Mansour. 2020. End-to-end slot alignment and recognition for cross-lingual NLU. *arXiv preprint arXiv:2004.14353* (2020).

[63] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. 2021. Self-supervised video object segmentation by motion grouping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7177–7188.

[64] Dongfang Yang, Haolin Zhang, Ekim Yurtsever, Keith A Redmill, and Ümit Özgüner. 2022. Predicting pedestrian crossing intention with feature fusion and spatio-temporal attention. *IEEE Transactions on Intelligent Vehicles* 7, 2 (2022), 221–230.

[65] Fanghua Ye, Jarana Manotumruksa, Qiang Zhang, Shenghui Li, and Emine Yilmaz. 2021. Slot self-attentive dialogue state tracking. In *Proceedings of the Web Conference 2021*. 1598–1608.

[66] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. 2019. Clevrer: Collision events for video representation and reasoning. *arXiv preprint arXiv:1910.01442* (2019).

[67] Xiaofeng Yuan, Lin Li, Yuri AW Shardt, Yalin Wang, and Chunhua Yang. 2020. Deep learning with spatiotemporal attention-based LSTM for industrial soft sensor model development. *IEEE Transactions on Industrial Electronics* 68, 5 (2020), 4404–4414.

[68] Chuhan Zhang, Ankush Gupta, and Andrew Zisserman. 2022. Is an Object-Centric Video Representation Beneficial for Transfer?. In *Proceedings of the Asian Conference on Computer Vision*. 1976–1994.

[69] Jing Zhang, Aiping Liu, Min Gao, Xiang Chen, Xu Zhang, and Xun Chen. 2020. ECG-based multi-class arrhythmia detection using spatio-temporal attention-based convolutional recurrent neural network. *Artificial Intelligence in Medicine* 106 (2020), 101856.

[70] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 586–595.

[71] Daniel Zoran, Rishabh Kabra, Alexander Lerchner, and Danilo J Rezende. 2021. Parts: Unsupervised segmentation with slots, attention and independence maximization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10439–10447.

Appendix

This supplementary material file provides the appendix section to the main article.

A Baselines

To validate the effectiveness of STATM, we chose three baselines which are state-of-the-art in object-centric video scene decomposition for improvement and comparison. All three models include a similar module: Slot Attention [36] followed by a predictor (a transformer encoder block [51]).

SAVi. The SAVi model [27] consists of five main components: encoder, decoder, slot initialization, corrector, and predictor. The encoder utilizes a convolutional neural network as a backbone to extract features from input video frames. The slot initialization is either a simple MLP (in the case of bounding boxes or center of mass coordinates) or a convolutional neural network (in the case of segmentation masks), responsible for initializing slots based on the conditioning information (bounding boxes, center of mass coordinates, or segmentation masks) in the first frame. The corrector employs Slot Attention [36] to update slot information based on visual features from the encoder. The predictor, a transformer encoder block [51], utilizes self-attention among the slots for prediction. The output of the predictor initializes the corrector at the next time step, ensuring consistent object tracking over time. Finally, the decoder uses a Spatial Broadcast Decoder [56] to generate RGB predictions of optical flow (or reconstructed frames) and an alpha mask.

For the SAVi model in perception, we chose the official implementation of SAVi with a small CNN as backbone, which is trained and evaluated on downscaled 64*64 frames. We initialized the slots in the first frame using bounding boxes as hints (slot initialization is a simple MLP). We selected optical flow as the target for training the model. During training, we adjusted the batch size to 32, while keeping other settings and parameters the same as SAVi-small.

SAVi++. SAVi++ [15] has a structure similar to SAVi. During the training of SAVi++, we adjusted the batch size to 32, the number of training steps to 100k, while keeping other parameters consistent with described in SAVi++. For SAVi++*, the hyperparameters are same as the official implementation.

STEVE. STEVE [45] is an unsupervised object-centric scene decomposition model. This baseline employs discrete VAE [20] for encoding and reconstructing input frames x_t and generating discrete targets for the transformer decoder. It uses a similar structure combined with the encoder, corrector, and predictor in SAVi, called the recurrent slot encoder, to decompose input video frames x_t into slots. The slot-transformer decoder uses the slots obtained from the recurrent slot encoder to learn to predict the sampling targets from discrete VAE by minimizing the cross-entropy loss. We make no modifications to the official implementation of STEVE.

SlotFormer. SlotFormer [60] is a transformer-based framework for object-centric visual simulation. It leverages slots extracted by upstream modules like SAVi to train a slot-based transformer encoder model for prediction purposes. Additionally, it utilizes results from rollout simulations to train Aloe [10] for Visual Question Answering (VQA) tasks. We employ an unsupervised approach to train SAVi-small on CLEVRER for slot extraction, with all other configurations consistent with the official settings.

SAVi-SlotFormer. To evaluate whether enhancements to the predictor can alter the performance of SAVi, we replaced the predictor in SAVi with SlotFormer, which is currently the best-performing model in object-centric prediction, featuring a memory buffer and transformer encoders predictor. All other settings are consistent with SAVi.

G-SWM. G-SWM [35] is an unsupervised, object-centric predictive model that calculates foreground and background distributions through two separate modules, subsequently rendering and combining these results for dynamic prediction. Object interactions and occlusions are managed through a simple graph neural network. The official implementation has been trained using CLEVRER, yielding results comparable to those obtained with SlotFormer [60].

SAVi-dyn. In SlotFormer [60], Wu et al. enhanced prediction capabilities by replacing the Transformer predictor in SAVi with a

Table S1: Comparison of the parameter number for different models.

Model	Parameter Number	Model	Parameter Number
SAVi-Small	895,268	STATM-SAVi-Small	961,572
SAVi-Medium	1,140,740	STATM-SAVi-Medium	1,207,044
SAVi-Large	22,273,412	STATM-SAVi-Large	22,339,716
SAVi++	23,132,165	STATM-SAVi++	23,264,389

Transformer-LSTM module in PARTS [71]. We trained SAVi-dyn using the same setup as theirs.

DCL. DCL [8] utilizes a trajectory extractor to monitor each object over time, representing it as a latent, object-centric feature vector. Building on this foundational representation, DCL employs graph networks to learn and approximate the dynamic interactions among objects.

Aloe. Aloe [10] trains transformers using slots for prediction. To enable a direct comparison with models such as SlotFormer, we use the Aloe model as re-implemented in the SlotFormer paper, ensuring all hyperparameters and settings are kept consistent with those described in the paper.

B Additional Training Setup

Experiments of STATM-SAVi and STATM-SAVi++. Referring to Section 4.1, we train our models (STATM-SAVi and STATM-SAVi++) for 100k steps with a batch size of 32 using Adam [26]. Same as SAVi++ [15], we linearly increase the learning rate for 5000 steps to 0.0002 (starting from 0) and then decay the learning rate with a Cosine schedule [37]. We split each video into sub-sequences of 6 frames to train the model (In the ablation study studying the impact of training stage buffer size on the model, sub-sequences are set to 4/6/12, referring to Section 4.3). In the initialization of slots for the first frame, bounding boxes are utilized as contextual cues. For MOVi-A, B, and C datasets, the number of slots is set to 11. In the case of datasets MOVi-D and E, the number of slots is set to 24. We use 1 iteration per frame for the Slot Attention [36] module. All other parameters and settings for each model remain consistent with their respective baselines. We implement models in JAX using the Flax neural network library, unless stated otherwise.

Experiments of STATM-SAVi++*. Further, we modify the training steps to 500k and change the batch size to 64 to train the STATM-SAVi++ model. Additionally, for the Slot Attention on the MOVi-E dataset, we adjust the number of iterations to 2 per frame. All other parameters and settings remain consistent with the SVAi++ [15].

Experiments of STATM-STEVE. Due to the necessity of memory in STATM, we modify the training subsequence length to 3/6/24 (corresponding to batch sizes of 24/12/8). During the experiments, we found that the STATM-STEVE model is sensitive to the MSE loss in discrete VAE [20], and the addition of STATM slightly increases the difficulty of fitting of the model, which becomes more pronounced with an increase in the buffer size of STATM module. Therefore, for a better improvement and testing of the impact of STATM on STEVE, a two-stage training approach can be attempted, where a discrete VAE is first trained, followed by the training of STATM and other modules. All other parameters and settings remain consistent with the STEVE [45]. We implement STATM-STEVE in PyTorch.

Experiments of Prediction and VQA. We train our models (STATM-SAVi) for 400k steps with a batch size of 64 on the CLEVRER dataset to extract slots. The number of slots is set to 7, with a learning rate of 0.0001. For prediction, we subsample the video by a factor of 2 to train STATM, conducting approximately 500k training steps with a batch size of 64 and a learning rate of 0.0002. We use rollout slots to train Aloe, targeting around 300k steps with a learning rate of 0.0001 and a batch size of 128. We use the Adam optimizer and apply the same warm-up and decay learning rate schedule for the first 2.5% of the total training steps.

C Additional Parameter

Using the STATM structure as a predictor does lead to a slight increase in the parameter count of the SAVi and SAVi++ models. However, under the same training settings, our model achieves superior metrics. This suggests that the moderate increase in the parameter count doesn't significantly increase the training complexity of our model.

STATM-SAVi-Small has a parameter increase of approximately 66K compared to SAVi-Small, which is notably smaller than the parameter increase seen in SAVi-Large compared to SAVi-Small (around 21378K parameters). Moreover, our STATM-SAVi-Small model, trained for 100k steps with a batch size of 32, performs similarly to the official SAVi-Large model, trained for 500k steps with a batch size of 64. This further highlights the reasonableness and superiority of our designed prediction module.

D Additional Experimental Results

Additional Metrics. All models were trained in a conditional setting, initializing slots using ground-truth bounding box information in the first frame. Consequently, the results for STATM-SAVi++* in Table 1 are measured from the second frame onward, aligning with the evaluation method in SAVi++ [15]. All other evaluation results include data from the first frame.

Additional Segmentation Results. In order to better assess our model, we conducted an evaluation using the first 6 frames of the videos. Detailed results can be found in Table S2. Referring to Table 1 in the main text, we can observe the following trends: on simple datasets, the decline in our model's object segmentation and tracking capabilities over extended time sequences is comparable to that of the baseline model. However, on complex datasets like MOVi-C, D, and E, the decrease in our model's performance is significantly less than that of the baseline model. This indicates that the STATM is more suitable for handling object segmentation and tracking tasks in longer-time sequences and complex environments. This finding further validates the effectiveness of our STATM model.

Table S2: Segmentation results on the first 6 frames of the MOVi dataset.

Model	mIoU↑ (%)					FG-ARI↑ (%)				
	A	B	C	D	E	A	B	C	D	E
SAVi-S	66.9	49.3	29.7	13.9	8.3	92.3	80.1	69.2	45.5	32.2
STATM-SAVi-S	71.0	51.6	43.5	21.9	12.5	92.6	81.7	73.0	50.2	54.7
SAVi++	85.2	59.5	55.3	49.8	30.7	97.2	86.3	83.9	87.1	88.2
STATM-SAVi++	85.8	59.8	56.8	56.7	31.1	97.2	86.6	83.9	89.2	88.6

Table S3: Accuracy on different questions and average results on CLEVRER.

Model	Descriptive	Explanatory		Predictive		Counterfactual		Average
		per opt.	per ques.	per opt.	per ques.	per opt.	per ques.	
DCL	90.70	89.58	82.82	90.52	82.03	80.38	46.52	75.52
VRDP	93.40	96.30	91.94	95.68	91.35	94.83	84.29	90.24
SlotFormer	95.17	98.04	94.79	96.50	93.29	90.63	73.78	89.26
STATM(Ours)	95.22	98.15	95.04	96.62	93.63	90.57	73.90	89.44

Additional Qualitative Results. We show more qualitative results on longer time series in Figure S1 to Figure S5. Meanwhile, To analyze slots and better illustrate the relationship between objects and slots, we visualized the attention map of the Slot Attention (corrector) in Figure S6 to Figure S10.

Additional VQA Results. Detailed results about all questions on CLEVRER are presented in Table S3.

E Additional Ablation Study

Training with unlimited buffer length and testing with limited buffer length. To better assess the impact of the buffer on the model, we trained the model using video sub-sequences of 12 frames, as shown in Table S4 and Table S5. We observed that: 1) On relatively simple datasets like MOVi-A, B, C, and D dataset, increasing the amount of training data with additional information would enhance the model's segmentation capabilities. 2) Training with a buffer size of 12 results in a decrease in mIoU. SAVi++ augments the number of samples by utilizing sub-sequences of 6 frames. However, in this case, sub-sequences of 12 frames are employed, leading to a reduction in both the number of samples and hints. The decrease in both samples and hints may impact the model's ability to separate foreground and background, consequently causing a decline in mIoU. 3) On the MOVi-E dataset, increasing the number of training frames resulted in a decrease in the model's tracking and segmentation capabilities. This could be attributed to the limitations in the ability of the upstream modules to effectively extract image features. The findings from the SAVi and SAVi++, which used more powerful encoders and data augmentation to improve segmentation performance on MOVi-E, support this observation. Therefore, exploring the design of a more robust encoder and refining the corrector and guidance modules may yield unexpected improvements. We plan to further investigate this direction in future research.

Training with limited buffer length and testing with limited buffer length. We intentionally limited the buffer size during both the training and testing phases, and the model evaluation results are presented in Table S6. Remarkably, we found that the model trained with a smaller buffer experienced less impact from the buffer during the testing phase. For instance, consider a model trained with a buffer size of 2. When tested with a reduced buffer size on the MOVi-E dataset, the model experienced an approximately 8% decrease in FG-ARI (from 23.6% to 15.5%). On the other hand, when testing with a reduced buffer size on the MOVi-E dataset, a model trained with a buffer size of 6 exhibited a FG-ARI decrease of about 30% (from 36.8% to 6.8%). This has intriguing implications for the fusion of deep learning and cognitive science. However, it's important to note that real human learning and cognitive processes are likely more complex and influenced by various factors. This study provides a theoretical framework, but further theoretical substantiation and experimental validation are still needed.

VQA results of different buffer size. Detailed results for all questions on CLEVRER using different memory buffer sizes are presented in Table S7.

F STATM Structure in Different Model

Due to the differences between the SAVi/SAVi++ and STEVE models, there are certain distinctions in the enhanced models' STATM module as well. Table S8 illustrates the simplified algorithm for the STATM module in different models.

The STATM in STATM-SAVi /SAVi++ employs post-normalization, with the residual structure applies to the last MLP layer of the module. On the other hand, the STATM in STATM-STEVE utilizes pre-normalization, where S_t and M_t share normalization weights, and the entire module applies a residual structure. The size of the key(k), query(q), and value(v) in the spatiotemporal attention for STATM-SAVi is 128, while in STATM-STEVE, it is 192.

Table S4: Evaluation on all video frames of the model trained using 12 frames (B represents the size of the buffer during the testing phase).

Model	mIoU↑ (%)					FG-ARI↑ (%)				
	A	B	C	D	E	A	B	C	D	E
STATM (Ball)	66.9	39.3	26.1	13.8	4.3	92.3	72.9	62.5	59.6	17.9
STATM (B12)	66.2	39.3	25.9	13.2	3.9	91.3	73.0	60.8	55.6	10.4
STATM (B6)	64.3	39.3	25.4	12.3	3.6	89.3	72.7	57.4	50.6	5.6
STATM (B4)	62.8	39.1	24.8	11.8	3.4	88.4	72.5	55.1	47.7	4.4
STATM (B2)	59.1	38.2	23.9	11.2	3.1	85.5	70.6	51.1	44.0	3.5

Table S5: Evaluation on the first 6 video frames of the models trained by 6 frames and 12 frames (T represents the number of frames used for training model, B represents the size of the buffer during the testing phase).

Model	mIoU↑ (%)					FG-ARI↑ (%)				
	A	B	C	D	E	A	B	C	D	E
STATM (T6, Ball)	71.0	51.6	43.5	21.9	12.5	92.6	81.7	73.0	50.2	54.7
STATM (T6, B6)	71.0	51.6	43.5	21.9	12.5	92.6	81.7	73.0	50.2	54.7
STATM (T6, B4)	71.0	51.3	42.7	19.7	12.0	92.6	81.7	72.5	46.8	51.0
STATM (T6, B2)	70.8	49.2	38.7	14.9	10.2	91.8	80.6	69.1	34.6	37.3
STATM (T12, Ball)	60.2	42.7	28.1	15.4	7.6	92.7	82.9	73.6	55.5	33.5
STATM (T12, B12)	60.2	42.7	28.1	15.4	7.6	92.7	82.9	73.6	55.5	33.5
STATM (T12, B6)	60.2	42.7	28.1	15.4	7.6	92.7	82.9	73.6	55.5	33.5
STATM (T12, B4)	59.7	42.8	28.0	15.4	7.3	92.1	82.9	73.3	54.5	29.3
STATM (T12, B2)	58.3	42.5	27.7	14.9	6.4	89.6	82.5	71.6	50.2	19.4

Table S6: Evaluation result of the model trained limited buffer length (T represents the size of the buffer during the training phase, B represents the size of the buffer during the testing phase).

Model	mIoU↑ (%)				FG-ARI↑ (%)			
	First 6 frames		All frames		First 6 frames		All frames	
	A	E	A	E	A	E	A	E
STATM (T2, Ball)	71.6	9.9	66.9	6.2	92.6	41.2	90.7	23.6
STATM (T2, B6)	71.6	9.9	69.0	5.7	92.6	41.2	91.5	22.7
STATM (T2, B4)	71.7	9.9	69.3	5.4	92.6	41.5	91.2	20.0
STATM (T2, B2)	71.9	9.5	69.5	4.8	92.6	41.3	91.2	15.5
STATM (T4, Ball)	73.6	9.8	68.0	6.8	92.5	41.7	90.4	30.1
STATM (T4, B6)	73.6	9.8	69.5	4.7	92.5	41.7	90.3	15.1
STATM (T4, B4)	73.7	9.7	69.6	4.3	92.4	41.3	90.1	11.8
STATM (T4, B2)	74.0	9.0	69.3	3.9	92.1	38.4	89.2	9.1
STATM (T6, Ball)	71.0	12.5	67.5	8.5	92.6	54.7	91.1	36.8
STATM (T6, B6)	71.0	12.5	66.1	5.4	92.6	54.7	89.8	12.8
STATM (T6, B4)	71.0	12.0	64.2	4.9	92.6	51.0	87.6	9.9
STATM (T6, B2)	70.8	10.2	61.6	4.2	91.8	37.3	85.5	6.8

Table S7: Accuracy on different questions and average results of different buffer sizes on CLEVRER.

Model	Descriptive	Explanatory		Predictive		Counterfactual		Average
		per opt.	per ques.	per opt.	per ques.	per opt.	per ques.	
STATM(24)	95.22	98.15	95.04	96.62	93.63	90.57	73.90	89.44
STATM(32)	95.34	98.23	95.30	96.41	93.09	90.91	74.69	89.61
STATM(48)	95.17	98.14	95.14	96.07	92.67	91.01	74.51	89.37
STATM(128)	95.15	97.84	94.15	96.21	92.90	90.50	73.44	88.91

Table S8: Simplified algorithm for the STATM module in different models.

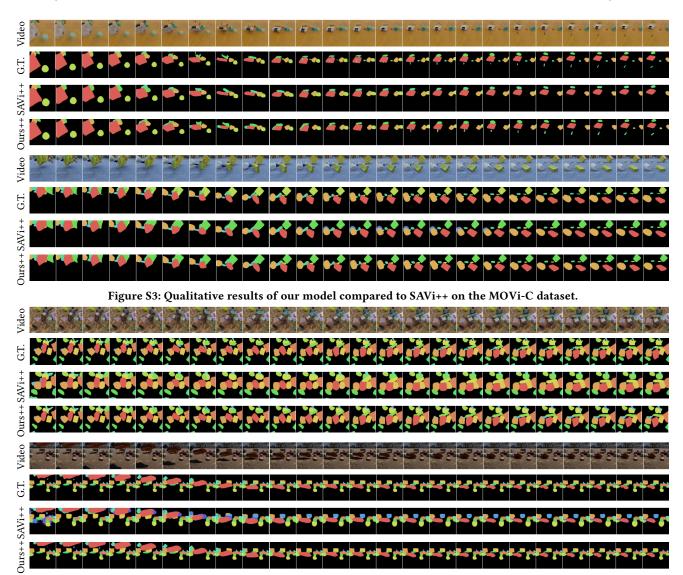| STATM in STATM-SAVi /SAVi++	STATM in STATM-STEVE
Input: S_t, M_t	**Input:** S_t, M_t
	$S_t = \text{LayerNorm}(S_t)$
	$M_t = \text{LayerNorm}(M_t)$
$X_t = \text{Spatiotemporal Attention}(S_t, M_t, M_t)$	$X_t = \text{Spatiotemporal Attention}(S_t, M_t, M_t)$
$X_t = \text{LayerNorm}(X_t + S_t)$	$X_t = \text{LayerNorm}(X_t + S_t)$
$Y_t = \text{MLP}(X_t)$	$Y_t = \text{MLP}(X_t)$
$Y_t = \text{LayerNorm}(Y_t)$	
Return: $X_t + Y_t$	**Return:** $S_t + Y_t$

Figure S1: Qualitative results of our model compared to SAVi++ on the MOVi-A dataset.

Figure S2: Qualitative results of our model compared to SAVi++ on the MOVi-B dataset.

Jian Li, Pu Ren, Yang Liu and Hao Sun

Figure S3: Qualitative results of our model compared to SAVi++ on the MOVi-C dataset.

Figure S4: Qualitative results of our model compared to SAVi++ on the MOVi-D dataset.

Figure S5: Qualitative results of our model compared to SAVi++ on the MOVi-E dataset.

frame   gt_fl   gt_sg   slot 1   slot 2   slot 3   slot 4   slot 5   slot 6   slot 7   slot 8   slot 9   slot 10   slot 11

Figure S6: Attention map visualization on the MOVi-A dataset.

Figure S7: Attention map visualization on the MOVi-B dataset.

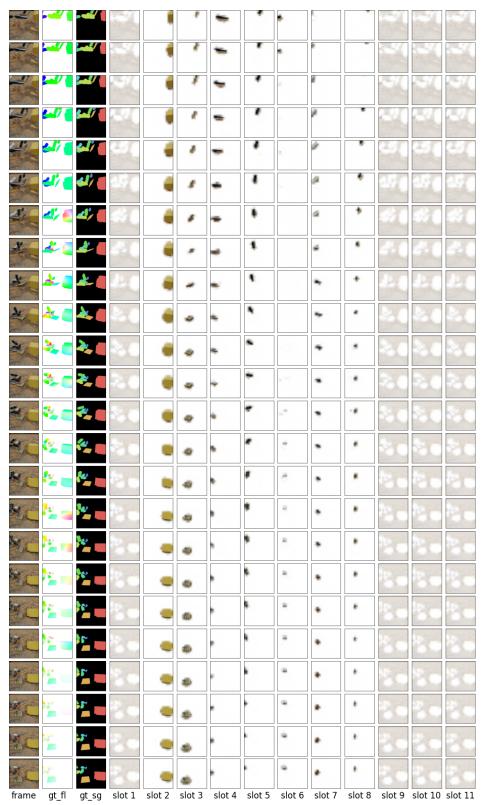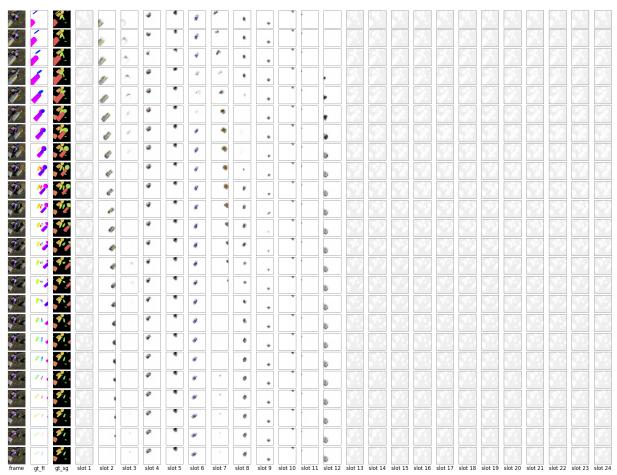**Figure S8: Attention map visualization on the MOVi-C dataset.**

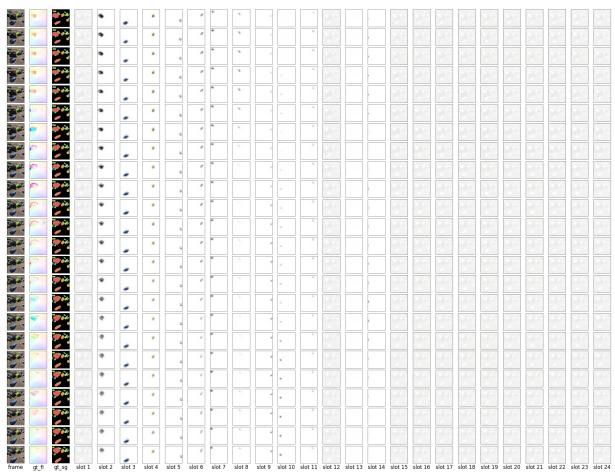**Figure S9: Attention map visualization on the MOVi-D dataset.**

Figure S10: Attention map visualization on the MOVi-E dataset.