Object-Centric Temporal Consistency via Conditional Autoregressive Inductive Biases

Cristian Meo[*,1] **Akihiro Nakano**[*,2]

Mircea Lică[1] **Aniket Didolkar**[3] **Masahiro Suzuki**[2] **Anirudh Goyal**[3]

Mengmi Zhang[4,5] **Justin Dauwels**[1] **Yutaka Matsuo**[2] **Yoshua Bengio**[3]

Abstract

Unsupervised object-centric learning from videos is a promising approach towards learning compositional representations that can be applied to various downstream tasks, such as prediction and reasoning. Recently, it was shown that pretrained Vision Transformers (ViTs) can be useful to learn object-centric representations on real-world video datasets. However, while these approaches succeed at extracting objects from the scenes, the slot-based representations fail to maintain temporal consistency across consecutive frames in a video, i.e. the mapping of objects to slots changes across the video. To address this, we introduce Conditional Autoregressive Slot Attention (CA-SA), a framework that enhances the temporal consistency of extracted object-centric representations in video-centric vision tasks. Leveraging an autoregressive prior network to condition representations on previous timesteps and a novel consistency loss function, CA-SA predicts future slot representations and imposes consistency across frames. We present qualitative and quantitative results showing that our proposed method outperforms the considered baselines on downstream tasks, such as video prediction and visual question-answering tasks.

1 Introduction

The main goal of object-centric (OC) representation learning is to represent each object in an image as a set of separate fixed-size vector representations called "slots" [1, 8, 10, 12, 37]. This slot-based representation serves to represent natural scenes as a composition of objects [12, 24, 37]. Due to this compositional nature of scenes [26], object-centric representations can enhance out-of-distribution generalization [12], and handle complex tasks such as reasoning [1, 39, 53, 54, 55], planning [40, 50], control [6, 38, 59], and reinforcement learning [10, 19, 58, 59]. Moreover, object-centric representation learning is in line with studies on the characterization of human perception and reasoning [29], making it a very appealing direction in terms of explainability [41] as well.

Although recent OC pipelines succeeds at accurately extracting objects from frames in a video [30, 54, 60], a persistent problem when applying object-centric models developed for images [37, 43, 44] to videos is temporal consistency. Although learning temporal consistent representations has been a central problem for many years [4, 14, 18, 20, 22, 23, 60], learning temporal consistent object-centric representations is particularly difficult as the representations are permutation-equivariant. Prior works utilize various architectural biases to achieve temporal consistency. Some approaches have explored

[*] Equal contributor. [1] Delft University of Technology, NL. [2] The University of Tokyo, JP. [3] Mila, University of Montreal, CA. [4] Deep NeuroCognition Lab, CFAR and I2R, Agency for Science, Technology and Research, SG. [5] Nanyang Technological University, SG.
Corresponding author: c.meo@tudelft.nl, nakano.akihiro@weblab.t.u-tokyo.ac.jp

NeurIPS 2024 Workshop on Compositional Learning: Perspectives, Methods, and Paths Forward.

employing prior networks to model temporal consistency expicitly [33, 45, 53, 54]. Other models have directly conditioned the slot representations on previous timesteps [21, 22, 45]. However, we argue that architectural biases may not always be enough to achieve temporal consistency. Another approach is to add an auxiliary loss in the representation space [60]. Contrary to Zadaianchuk et al. [60], we argue that adding such a loss directly on the slots encourages the representations to be too similar between timesteps, which may hinder the model's ability to generalize to longer sequences.

To mitigate the problem of temporal consistency, we propose Conditional Autoregressive Slot Attention (CA-SA), a model-agnostic module that consists of: (1) An autoregressive network that predicts the initial slot representations of the current timestep from the previous timestep, to condition the current slot extraction on prior timesteps, and (2) A temporal consistency loss between the feature-to-slots attention maps of two consecutive frames, to impose the same slot to attend to spatially similar area of the image. Through ablations, we show that the combination of the two is the key to learning a more temporally consistent representations. We present qualitative and quantitative evaluations of the proposed approach on the CLEVRER [57] and Physion [3] datasets, showing how objects' temporal consistency improve in terms of downstream task performance.

2 Related Works

The problem of temporal inconsistency has been studied for many years [5]. Whenever various image processing algorithms are applied as precursors to video processing, certain temporal inconsistencies can be introduced in the consecutive frames of the video. For example, certain noise reduction algorithms may cause flickering due to slight variations in noise patterns of consecutive frames. To deal with such inconsistencies, previous works have introduced various objectives and priors. Lai et al. [34] introduces a perceptual loss to encourage temporal consistency. Eilertsen et al. [15], introduce two regularization terms that force a frame and its affine transformation to have similar representations. A range of approaches also rely on predicting optical flow or motion information for achieving temporal consistency [7, 13, 35, 56]. While these works consider general computer-vision problems, the importance of temporal consistency also applies to video-based object-centric models as well. To ensure temporal consistency various approaches replace the sampling operation, which introduces the permutation equivariance property of slots [37], by conditioning slots on previous ones [22, 33, 45, 53]. In this work, besides introducing a novel architectural bias, we introduce an auxiliary loss which enforces consistency by optimizing for consecutive attention maps to be similar.

Related works on object discovery and video downstream tasks are summarized in Appendix B.

3 Method

When it comes to modelling sequences with an autoregressive model (e.g., RNN [16], autoregressive transformer [49]), ensuring objects-to-slot consistency is necessary to learn meaningful objects dynamics [22, 23, 24]. In contrast to most existing methods, which either enforce an architectural bias [17, 33, 45, 53] or a regularizer to enhance temporally consistent object slots [10, 60], our method proposes to use both. Table 6 shows a comparison of our proposed method with existing approaches which try to mitigate permutation equivariance property of object-centric representations.

In this section, we present the two main contributions of our proposed approach, namely, (1) CA-SA (Conditional Autoregressive Slot Attention), an autoregressive network that predicts the initial slot representations of the consecutive next timestep and conditions the current slot extraction on prior timesteps, and (2) OPC (Objects Permutation Consistency Loss), an auxiliary loss between two consecutive attention score matrices of the feature-to-slots attention mechanisms, to impose objects permutation temporal consistency between different timesteps. Our proposed objective and architecture are shown in Figure 1. As our method is architecture-agnostic, this makes it suitable for any SA-based model for videos.

The overall pipeline, frame generation procedure, and preliminaries about Slot Attention [37] can be found in Appendix A.

3.1 CA-SA: Conditional Autoregressive Slot Attention

Conditional Autoregressive Prior. Given an input video consisting of T frames $x_{1:T}$, each input image is first individually encoded via a feature extractor to latent features $z_{1:T}$. Then, features are fed into the Slot Attention architecture to infer slot representations $s_{1:T}^{1:K}$. Since our goal is to model

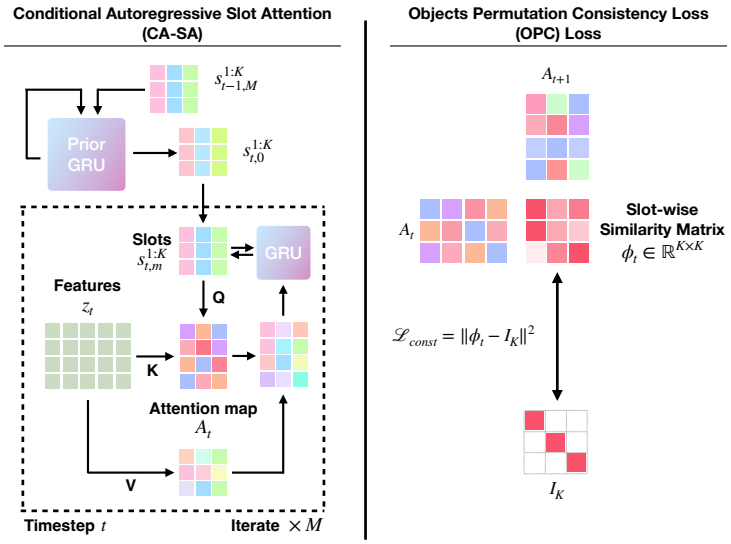

Figure 1: Left: Overall CA-SA architecture is represented. The Prior GRU network takes the slots from the previous timestep and condition the initialization of the new slots. The vanilla SA is represented within the dashed box. Right: Visualization of the OPC loss. Two consecutive attention maps A_t, A_{t+1} are used to compute a cosine similarity distance, whose diagonal elements are optimized to match an identity matrix to impose slots' temporal consistency.

an object-centric dynamics of the environment using slot representations, we need to ensure that the same slots are used to represent the same objects in the scene along the whole video trajectory. In this work, we empirically find that updating individual slots via a Gated Recurrent unit (GRU) based prior network yields the best results:

$$\tilde{s}_t^k, h_t = \text{GRU}_{\text{prior}}(s_{t-1}^k, h_{t-1}) \ , \ k = [1, 2, \cdots, K], \tag{1}$$

where $s_t, \tilde{s}_t, h_t$ are the slots, initial slots, and the hidden state of the $\text{GRU}_{\text{prior}}$, respectively. t denotes the timestep. Unlike previous conditioning approaches, which allow for inter-slot interaction using MLP or Transformer, the GRU prior network imposes a structure that prevents representation mixing and preserves the object identity.

OPC: Objects Permutation Consistency Loss. To define a meaningful consistency loss, we draw inspiration from Spelke [46]'s findings of how human infants perceive objects using several properties, one of which being their spatiotemporal continuity. For Slot Attention, this principle can be translated into the notion that attention maps generated at consecutive timesteps should exhibit consistency, reflecting the assumption that the same object would persist in spatially proximate pixels across successive frames [9]. However, when defining such consistency within slots, the imposed inductive bias results to be too strong. Indeed, as the loss is backpropagated backward in time through the prior network of slot representations, the cumulative effect of minor alterations in the representations can lead to their deterioration [40].

To solve this issue, we focus on the attention map that is computed within the Slot Attention architecture per timestep. Using attention maps allows us to define a weaker regularization, which does not compromise the slot representation while ensuring slot permutation consistency. Formally, let the attention map at timestep t as $A_t \in \mathbb{R}^{K \times H'W'}$. Given attention maps at consecutive timesteps, A_t, A_{t+1}, to encourage the attention maps for the same slot to be consistent over different timesteps, we define OPC as:

$$\mathcal{L}_{\text{OPC}} = \frac{1}{TK} \sum_{t=1}^{T} \sum_{i=1}^{K} \|(\phi_t - I_K)_{ii}\|^2 \ , \ \phi_t = \frac{A_t A_{t+1}^{\text{T}}}{\|A_t\| \|A_{t+1}\|}, \tag{2}$$

where ϕ_t is the attention-wise cosine similarity between consecutive attention maps and $I_K \in \mathbb{R}^{K \times K}$ is an identity matrix. Overall, the proposed method is model-agnostic to any slot-based object-centric

Figure 2: Generation results and predicted masks on CLEVRER (above) and Physion (below). Red square indicate slots which temporal consistency is improved by adding CA-SA.

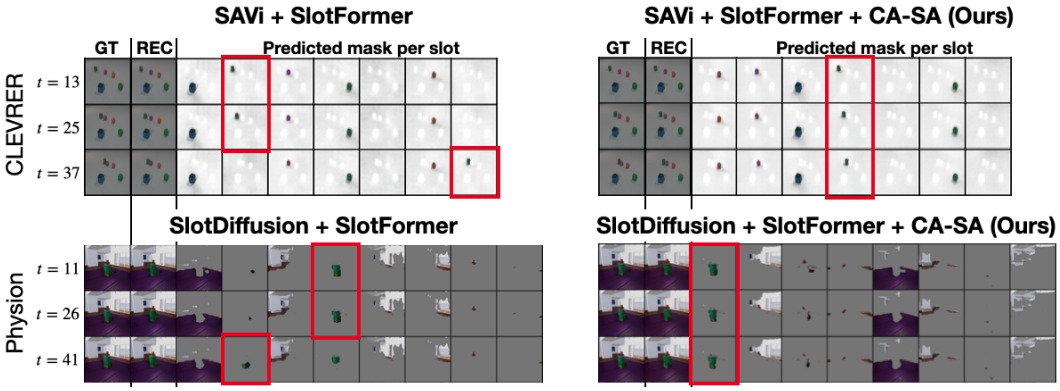

Table 1: Evaluation of video prediction task on CLEVRER dataset. * indicates reproduced results. Best results are indicated in **bold**.

Model	Visual quality			Object dynamics			
	PSNR (↑)	SSIM (↑)	LPIPS (↓)	AR % (↑)	ARI % (↑)	FG-ARI %(↑)	FG-mIoU % (↑)
SAVi-dyn	**29.77**	**0.89**	0.19	8.94	8.64	64.32	18.25
SAVi + SlotFormer*	29.22	0.87	0.15	44.19	58.49	65.96	27.90
SAVi + SlotFormer + CA-SA(Ours)	29.47	0.88	**0.14**	**46.50**	**60.52**	**67.25**	**28.60**

learning approach for videos. To incorporate our method, we add the object permutation consistency objective to the original loss function of the method that we wish to apply to. In our case, we optimize the OC feature extractor with a spatial broadcast decoder (SBD) [37, 51] to reconstruct the images from slots. The model is trained using an image reconstruction loss as in [33] together with our proposed objective:

$$\mathcal{L}_{\text{OC-feature extractor}} = \mathcal{L}_{\text{image}} + \lambda \mathcal{L}_{\text{OPC}} \ , \ \mathcal{L}_{\text{image}} = \text{MSE}(x_{1:T}, \hat{x}_{1:T}) \tag{3}$$

where λ is a hyperparameter. In our experiments, we set the value to $\lambda = 0.1$ for all datasets.

4 Experiments

We conduct experiments to evaluate CA-SA by exploring the following question: Do temporally consistent object-centric representations improve their downstream usefulness on video-related tasks? To answer this question, we validate the proposed model on video prediction (VP) and visual question answering (VQA) using CLEVRER [57] and Physion [3] datasets. We also conduct ablational experiments in subsection 4.3. We provide further details on the datasets and experimental setup in Appendix C and Appendix D, respectively.

4.1 Video Prediction Task

Table 1 and Table 2 show the results of the video prediction task for CLEVRER and Physion dataset, respectively. Figure 2 shows examples of generated slots for both datasets. On CLEVRER dataset, as the table shows, CA-SA outperforms other baseline models both in terms of visual quality and object-level segmentation. Our model is competitive in terms of visual quality, as the image encoder is the same as in the baseline model. We see that adding temporal consistency improves object-level segmentation for all metrics. On Physion dataset, according to Table 2 the proposed model performs

Table 2: Evaluation of video prediction task on Physion dataset. * indicates reproduced results. Best results are indicated in **bold**.

Model	MSE (↓)	LPIPS (↓)	FVD (↓)
STEVE + SlotFormer	832.0	0.43	930.6
SlotDiffusion + SlotFormer*	**489.5**	**0.27**	**737.8**
SlotDiffusion + SlotFormer + CA-SA(Ours)	502.6	**0.27**	759.0

Table 3: VQA task on CLEVRER [57], reporting per-option (per opt.) and per-question (per ques.) accuracy. SF stands for SlotFormer. Both models use Aloe to perform the VQA task. * indicates reproduced results, best ones are in **bold**.

Model	per opt. (%)	per ques. (%)
SF + Aloe*	90.72	80.22
SF + Aloe + CA-SA (Ours)	**92.69**	**84.88**

Table 4: VQA task on Physion [3], reporting accuracy on burn-in (Obs.) and burn-in plus rollout frames (Dyn.). SD and SF stand for SlotDiffusion and SlotFormer, respectively. * indicates reproduced results, best ones are in **bold**.

Model	Obs. (%)	Dyn. (%)
SD + SF*	63.8	63.9
SD + SF + CA-SA (Ours)	**64.1**	**64.7**

Table 5: Ablation study on video object discovery task of CLEVRER dataset.

Model	Prior	Aux. Loss	Visual quality			Object dynamics			
			PSNR (↑)	SSIM (↑)	LPIPS (↓)	AR (↑)	ARI (↑)	FG-ARI (↑)	FG-mIoU (↑)
CA-SA	✔ (GRU)	✔	**40.67**	**0.98**	**0.07**	**78.98**	79.19	**93.94**	**40.71**
CA-SA w/o prior	✗	✔	39.32	0.97	0.08	76.28	79.15	93.83	39.54
CA-SA w/o aux. loss	✔ (GRU)	✗	38.92	0.97	0.08	38.12	61.28	93.60	35.32
StoSAVi	✔ (MLP)	✗	39.81	0.97	0.08	80.47	**79.44**	93.91	40.51

slightly worse than SlotDiffusion + SlotFormer for MSE and FVD, while tying for LPIPS with a value of 0.27.

The performance disparity among datasets can be attributed to their distinct characteristics. Most object-centric models are trained with a surplus of slots compared to the total number of objects in the scene [12, 37]. As CLEVRER dataset exhibits a simpler background than Physion, this potentially results in disentanglement with multiple "empty" slots, i.e. slots which attend to neither the foreground objects nor the background [53]. Consequently, models trained on CLEVRER show greater performance enhancements over baseline models due to the possibility of temporal inconsistencies arising from empty slots, whereas achieving temporal consistency is more straightforward on Physion.

4.2 Video Question Answering Task

Table 3 and Table 4 summarize the results on CLEVRER and Physion VQA tasks, respectively. On CLEVER dataset, adding our proposed method improves the VQA accuracy by 1.9% and 4.6% for accuracy per-option and per-question, respectively. On Physion dataset, our model slightly improves accuracy of both metrics. The detailed results of both datasets are in Appendix F. Again, we observe that the performance gain is larger for CLEVRER dataset, as our model is able to reduce the temporal inconsistency caused by empty slots.

4.3 Ablation Study

In this section, we provide ablation results on model architecture of CA-SA. We report visual quality and object dynamics in video object discovery task of CLEVRER dataset in Table 5. As the result shows, the combination of using a GRU prior and the proposed auxiliary loss improves over vanilla stochastic SAVi in all metrics except for ARI.

5 Conclusion

In this paper, we proposed CA-SA, a model-agnostic module consisting an autoregressive network and OPC, an auxiliary loss aimed to improve object-to-slot temporal consistency of video object-centric models. We experimented on two types of downstream tasks, VP and VQA, and showed that adding our proposed method on top of state-of-the-art baselines improve their performances. Particularly, while we observed a marginal improvement in the video prediction task, CA-SA enhanced the VQA downstream performance across all metrics. We justified such difference considering that, while the VP task relies on the image space, VQA task uses the extracted slots as input space, clearly showing the importance of having temporal consistent slots. As in Slotformer [53], we observed that the two-stage training strategy harms the model's performance at the early rollout steps. Exploring joint training of the base object-centric model and the Transformer dynamics module could potentially benefit the performance of both models. We also leave investigation of combining our method with other video object-centric models and applying our method on wider variation of downstream tasks as future works.

References

[1] R. Assouel, P. Rodriguez, P. Taslakian, D. Vazquez, and Y. Bengio. Object-centric compositional imagination for visual abstract reasoning. In *ICLR2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality*, 2022. URL `https://openreview.net/forum?id=rCzfIruU5x5`.

[2] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] D. Bear, E. Wang, D. Mrowca, F. Binder, H.-Y. Tung, P. RT, C. Holdaway, S. Tao, K. Smith, F.-Y. Sun, F.-F. Li, N. Kanwisher, J. Tenenbaum, D. Yamins, and J. Fan. Physion: Evaluating physical prediction from vision in humans and machines. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

[4] Y. Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 17–36. JMLR Workshop and Conference Proceedings, 2012.

[5] Y. Bengio and J. Bergstra. Slow, decorrelated features for pretraining complex cell-like networks. *Advances in neural information processing systems*, 22, 2009.

[6] O. Biza, R. Platt, J.-W. van de Meent, L. L. Wong, and T. Kipf. Binding actions to objects in world models. In *ICLR2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality*, 2022.

[7] N. Bonneel, J. Tompkin, K. Sunkavalli, D. Sun, S. Paris, and H. Pfister. Blind video temporal consistency. *ACM Transactions on Graphics (TOG)*, 34(6):1–9, 2015.

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

[9] A. Chakravarthy, T. Nguyen, A. Goyal, Y. Bengio, and M. C. Mozer. Spotlight attention: Robust object-centric learning with a spatial locality prior. *arXiv preprint arXiv:2305.19550*, 2023.

[10] A. Didolkar, A. Goyal, and Y. Bengio. Cycle consistency driven object discovery. In *International Conference on Learning Representations*, 2024.

[11] D. Ding, F. Hill, A. Santoro, M. Reynolds, and M. Botvinick. Attention over learned object embeddings enables complex visual reasoning. *Advances in neural information processing systems*, 34:9112–9124, 2021.

[12] A. Dittadi, S. Papa, M. De Vita, B. Schölkopf, O. Winther, and F. Locatello. Generalization and robustness implications in object-centric learning. *arXiv preprint arXiv:2107.00637*, 2021.

[13] X. Dong, B. Bonev, Y. Zhu, and A. L. Yuille. Region-based temporally consistent video post-processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 714–722, 2015.

[14] A. A. Duval, V. Schmidt, A. Hernández-García, S. Miret, F. D. Malliaros, Y. Bengio, and D. Rolnick. FAENet: Frame averaging equivariant GNN for materials modeling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 9013–9033. PMLR, 2023.

[15] G. Eilertsen, R. K. Mantiuk, and J. Unger. Single-frame regularization for temporally stable cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11176–11185, 2019.

[16] J. L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990. ISSN 0364-0213. doi: https://doi.org/10.1016/0364-0213(90)90002-E. URL `https://www.sciencedirect.com/science/article/pii/036402139090002E`.

[17] G. Elsayed, A. Mahendran, S. Van Steenkiste, K. Greff, M. C. Mozer, and T. Kipf. Savi++: Towards end-to-end object-centric learning from real-world videos. *Advances in Neural Information Processing Systems*, 35:28940–28954, 2022.

[18] D. Erhan, A. Courville, and Y. Bengio. Understanding representations learned in deep architectures. *Department dInformatique et Recherche Operationnelle, University of Montreal, QC, Canada, Tech. Rep*, 1355(1):69, 2010.

[19] S. Ferraro, P. Mazzaglia, T. Verbelen, and B. Dhoedt. FOCUS: Object-centric world models for robotic manipulation. In *Intrinsically-Motivated and Open-Ended Learning Workshop @NeurIPS2023*, 2023. URL https://openreview.net/forum?id=RoQbZRv1zw.

[20] X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 513–520, 2011.

[21] A. Goyal, A. Lamb, P. Gampa, P. Beaudoin, S. Levine, C. Blundell, Y. Bengio, and M. Mozer. Object files and schemata: Factorizing declarative and procedural knowledge in dynamical systems. *arXiv preprint arXiv:2006.16225*, 2020.

[22] A. Goyal, A. Didolkar, N. R. Ke, C. Blundell, P. Beaudoin, N. Heess, M. C. Mozer, and Y. Bengio. Neural production systems. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 25673–25687. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/d785bf9067f8af9e078b93cf26de2b54-Paper.pdf.

[23] A. Goyal, A. Lamb, P. Gampa, P. Beaudoin, C. Blundell, S. Levine, Y. Bengio, and M. C. Mozer. Factorizing declarative and procedural knowledge in structured, dynamical environments. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=VVdmjgu7pKM.

[24] A. Goyal, A. Lamb, J. Hoffmann, S. Sodhani, S. Levine, Y. Bengio, and B. Schölkopf. Recurrent independent mechanisms. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=mLcmdlEUxy-.

[25] K. Greff, R. L. Kaufman, R. Kabra, N. Watters, C. Burgess, D. Zoran, L. Matthey, M. Botvinick, and A. Lerchner. Multi-object representation learning with iterative variational inference. In *International conference on machine learning*, pages 2424–2433. PMLR, 2019.

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

[27] B. Jia, Y. Liu, and S. Huang. Improving object-centric learning with query optimization. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=_-FN9mJsgg.

[28] J. Johnson, B. Hariharan, L. Van Der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017.

[29] D. Kahneman, A. Treisman, and B. J. Gibbs. The reviewing of object files: Object-specific integration of information. *Cognitive Psychology*, 24(2):175–219, 1992. ISSN 0010-0285. doi: https://doi.org/10.1016/0010-0285(92)90007-O. URL https://www.sciencedirect.com/science/article/pii/001002859290007O.

[30] I. Kakogeorgiou, S. Gidaris, K. Karantzalos, and N. Komodakis. Spot: Self-training with patch-order permutation for object-centric learning with autoregressive transformers, 2023.

[31] N. R. Ke, A. Didolkar, S. Mittal, A. G. ALIAS PARTH GOYAL, G. Lajoie, S. Bauer, D. Jimenez Rezende, M. Mozer, Y. Bengio, and C. Pal. Systematic evaluation of causal discovery in visual model based reinforcement learning. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

[32] T. Kipf, E. van der Pol, and M. Welling. Contrastive learning of structured world models. In *International Conference on Learning Representations*, 2019.

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

[34] W.-S. Lai, J.-B. Huang, O. Wang, E. Shechtman, E. Yumer, and M.-H. Yang. Learning blind video temporal consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 170–185, 2018.

[35] M. Lang, O. Wang, T. Aydin, A. Smolic, and M. Gross. Practical temporal consistency for image-based graphics applications. *ACM Transactions on Graphics (ToG)*, 31(4):1–8, 2012.

[36] Z. Lin, Y.-F. Wu, S. Peri, B. Fu, J. Jiang, and S. Ahn. Improving generative imagination in object-centric world models. In *International Conference on Machine Learning*, pages 6140–6149. PMLR, 2020.

[37] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf. Object-centric learning with slot attention. *Advances in neural information processing systems*, 33:11525–11538, 2020.

[38] D. Mambelli, F. Träuble, S. Bauer, B. Schölkopf, and F. Locatello. Compositional multi-object reinforcement learning with linear relation networks. In *ICLR2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality*, 2022.

[39] A. Mansouri, J. Hartford, Y. Zhang, and Y. Bengio. Object centric architectures enable efficient causal representation learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=r9FsiXZxZt`.

[40] A. Nakano, M. Suzuki, and Y. Matsuo. Interaction-based disentanglement of entities for object-centric world models. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=JQc2VowqCzz`.

[41] L. Pantelis, P. Vasilis, and K. Sotiris. Explainable ai: A review of machine learning interpretability methods. In *Entropy*, 2020.

[42] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[43] M. Seitzer, M. Horn, A. Zadaianchuk, D. Zietlow, T. Xiao, C.-J. Simon-Gabriel, T. He, Z. Zhang, B. Schölkopf, T. Brox, et al. Bridging the gap to real-world object-centric learning. In *International Conference on Learning Representations*, 2023.

[44] G. Singh, F. Deng, and S. Ahn. Illiterate dall-e learns to compose. In *International Conference on Learning Representations*, 2022.

[45] G. Singh, Y.-F. Wu, and S. Ahn. Simple unsupervised object-centric learning for complex and naturalistic videos. *Advances in Neural Information Processing Systems*, 35:18181–18196, 2022.

[46] E. S. Spelke. Where perceiving ends and thinking begins: The apprehension of objects in infancy. In *Perceptual development in infancy*, pages 197–234. Psychology Press, 2013.

[47] T. Unterthiner, S. van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly. FVD: A new metric for video generation, 2019. URL `https://openreview.net/forum?id=rylgEULtdN`.

[48] A. Van Den Oord, O. Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

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

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

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

[54] Z. Wu, J. Hu, W. Lu, I. Gilitschenski, and A. Garg. Slotdiffusion: Object-centric generative modeling with diffusion models. *Advances in Neural Information Processing Systems*, 36: 50932–50958, 2023.

[55] J. Yang, J. Mao, J. Wu, D. Parikh, D. D. Cox, J. B. Tenenbaum, and C. Gan. Object-centric diagnosis of visual reasoning. *arXiv preprint arXiv:2012.11587*, 2020.

[56] C.-H. Yao, C.-Y. Chang, and S.-Y. Chien. Occlusion-aware video temporal consistency. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 777–785, 2017.

[57] K. Yi*, C. Gan*, Y. Li, P. Kohli, J. Wu, A. Torralba, and J. B. Tenenbaum. Clevrer: Collision events for video representation and reasoning. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HkxYzANYDB`.

[58] J. Yoon, Y.-F. Wu, H. Bae, and S. Ahn. An investigation into pre-training object-centric representations for reinforcement learning. In *International Conference on Machine Learning*, pages 40147–40174. PMLR, 2023.

[59] A. Zadaianchuk, M. Seitzer, and G. Martius. Self-supervised visual reinforcement learning with object-centric representations. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=xppLmXCbOw1`.

[60] A. Zadaianchuk, M. Seitzer, and G. Martius. Object-centric learning for real-world videos by predicting temporal feature similarities. *Advances in Neural Information Processing Systems*, 36, 2023.

A Overall Pipeline

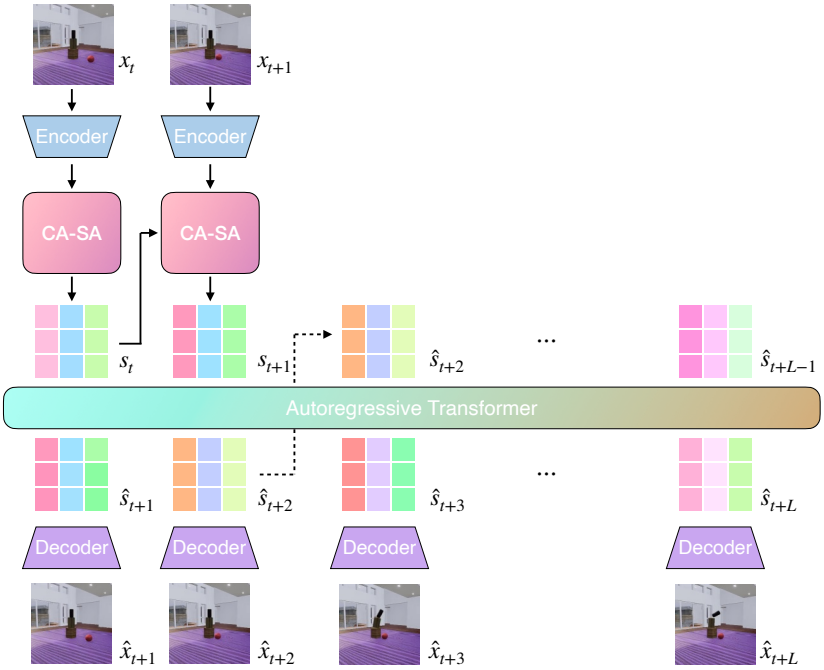

Figure 3: Proposed pipeline: Images x_t are first encoded into features, which are used to extract slots s_t. Slots video trajectory is generated using an autoregressive transformer and decoded into the predicted video using a Spatial Broadcast Decoder.

This section presents the overall pipeline, slot attention preliminaries, and frame generation procedure. Figure 3 shows the overall pipeline of CA-SA, where CA-SA is used to extract temporal consistent slots from the frames, and the autoregressive transformer is used to generate future slots.

A.1 Preliminaries

Slot Attention (SA). Slot Attention is an architecture proposed for unsupervised object-centric representation learning from images. An input image $x \in \mathbb{R}^{3\times H\times W}$ is processed through a Convolutional Neural Network (CNN) encoder (feature extractor) to extract features $z \in \mathbb{R}^{D_{enc}\times H'\times W'}$. Here, D_{enc} is the feature dimension, and H, W and H', W' are the height and width of the input and encoded image, respectively. The features are then combined with positional embeddings and flattened spatially. Then, the model initializes K, D_{slot}-dimensional object-centric representations, $\tilde{s}^{1:K} \in \mathbb{R}^{K\times D_{slot}}$, from some distribution. Using the slots as query and the features as key and value, Scaled Dot-Product Attention [49] is calculated for M iterations to update slot representations, $s^{1:K} = f_{SA}(\tilde{s}^{1:K}, M)$, where f_{SA} is the SA function. Unsupervised scene decomposition into individual objects is encouraged through the calculation of iterative self-attention by motivating the slots compete against each other to attend to different parts of the image. The slots are then fed to a spatial broadcast decoder (SBD) [51] to reconstruct the input image. The entire architecture is trained using image reconstruction loss only.

A.2 Frame Generation using Slot Representations

Given slot representations $s^{1:K}_{1:T}$, we wish to generate a sequence of future slots of length L, $\hat{s}^{1:K}_{T+1:T+L}$. To do so, we follow the approach proposed by SlotFormer [53] and employ an autoregressive transformer $\mathcal{T}$ [49] architecture to perform sequence modelling of the extracted slots. The autoregressive transformer input space is defined using a Multi-Layer Perceptron (MLP) layer, MLP_{in} which maps slots to embeddings, and positional encodings that are summed to the transformer embeddings to

Table 6: This table serves to highlight the differences of our models with prior works. The column Temporal-Consistency Prior indicates the approach taken by each model to ensure temporally consistent slot representations across frames in a video.

Method	Temporal-Consistency Prior			Tasks		
	Conditioning of slots	Auxiliary loss	Use RGB inputs only	Reconstruction	Prediction	VQA
SCOFF [21]	previous slots	✗	✓	✓	✓	✓
NPS [22]	previous slots	✗	✓	✓	✓	✗
STEVE [45]	previous slots	✗	✓	✓	✓	✗
SAVi [33]	Transformer prior	✗	✗	✓	✓	✗
VideoSAUR [60]	✗	✓	✓	✓	✗	✗
SlotFormer [53]	MLP/Transformer prior	✗	✓	✓	✓	✓
SlotDiffusion [54]	Transformer prior	✗	✓	✓	✓	✓
Ours	GRU prior	✓	✓	✓	✓	✓

impose a temporal structure. A MLP head MLP_{out} is used to map the transformer outputs back to the slot space. Overall, the sequence modelling can be formally expressed as,

$$u_{1:T}^{1:K} = \text{MLP}_{in}(s_{1:T}^{1:K}) \ , \ v_{1:T}^{1:K} = \mathcal{T}(\tilde{u}_{1:T}^{1:K}) \ , \ \hat{s}_{2:T+1}^{1:K} = \text{MLP}_{out}(v_{1:T}^{1:K}), \tag{4}$$

where $\tilde{u}_{1:T}^{1:K} = u_{1:T}^{1:K} + p_t$, p_t are the positional encodings [49] and each slot representation is used to predict the same slot at the next timestep. To generate a full trajectory, each slot is generated autoregressively following the approach defined by Wu et al. [53]. The autoregressive transformer $\mathcal{T}$ is optimized using the following autoregressive objective:

$$\mathcal{L}_{\text{Dyn}} = \text{MSE}(s_{2:T}^{1:K}, \hat{s}_{1:T-1}^{1:K}) + \text{MSE}(x_{2:T}^{1:K}, \hat{x}_{1:T-1}^{1:K}), \tag{5}$$

We use the SBD that was trained in section 3.1 to decode the predicted slots to image space. While training the autoregressive transformer, the weights of the SBD are kept frozen.

B Extended Related Works

Object-centric learning has been gathering attention as a promising direction towards learning efficient and compositional representations of complex scenes without supervision [10, 25, 27, 37, 44, 52]. While recent works have succeeded in applying this approach to real-world scenes [30, 43, 54, 60], their evaluation is limited to mask-based metrics. Contrary to this, this work focuses on evaluating the quality of the object-centric representations themselves by applying the representations to downstream tasks. Specifically, we focus on two types of downstream tasks - video prediction and visual question answering. While relatively few, there have been some works that have tackled problems similar to the ones we consider here [22, 31, 32, 50, 53, 54]. All of these works, except [53, 54], employ a factored representation coupled with a recurrent dynamics module for video prediction or world modelling. Wu et al. [53] and Wu et al. [54] adopt a transformer-based dynamics module. Out of this, [22, 53, 54] consider Slot Attention [37] as the base model to extract slots from the model. These models rely on architectural priors to impose temporally consistency between slots from neighbouring timesteps. Contrary to these methods, our approach introduces an objective that explicitly optimizes for temporal consistency. Moreover, our approach can be integrated into any of the above three approaches. Table 6 further highlights the differences between the proposed and existing approaches.

C Dataset Details

We validate the proposed model on the Video Prediction and Visual Question Answering downstream tasks on CLEVRER and Physion datasets. In this section, we provide further details on the dataset and preprocessing of the data.

CLEVRER. CLEVRER [57] consists of realistically rendered sequences with multiple 3D objects moving in the scene. The objects differ in shape, color, and texture. The size of each object are kept identical so that no vertical bouncing occurs during collision. The dataset, similar to CLEVR [28] and OBJ3D [36], features smaller objects and more diverse interactions of objects, making it a more challenging task. The attributes of the objects are randomly sampled under the constraint that none of

the objects in the scene have the identical attributes. Objects' positions are randomly initialized for each sequence. For each sequence, some objects are randomly chosen such that they cause a collision with each other. The dataset is accompanied by a VQA task with four types of questions: descriptive, explanatory, predictive, and counterfactual. Descriptive questions focus on understanding the video's dynamic content and temporal relations, asking about objects' attributes in an open-ended format. Explanatory questions explore causal relationships, asking which objects or events are responsible for other events. Predictive questions test the ability to predict future events. Counterfactual questions evaluate the understanding of hypothetical scenarios by asking what would or would not happen under altered conditions. Descriptive questions are open-ended questions, while the other three questions are in multiple-choice format with more than one possible answer.

Physion. Physion [3] consists of eight video categories, each showing a different physical phenomenon, such as rigid- and soft-body collisions, falling, rolling, and sliding motions. Each video category presents foreground objects, which vary in categories, textures, colors, and sizes, and diverse background scenes environment showed from randomized camera poses.

The Physion dataset consists of three set: Training, Readout Fitting, and Testing. Following SlotDiffusion [54], we sub-sample the frames by a factor of 3 for training the dynamics module and truncated by 150 frames, since that is the threshold within most of interactions happen. To validate models performances we adopt the official evaluation protocol. First, the dynamics models are trained on videos from the Training set. Then, conditioned by the first 45 frames of Readout Fitting and Test videos, they perform rollout to generate future scene representations. A linear readout model is trained on observed and rollout scene representations from the Readout Fitting set to classify whether an "agent" object (colored in red) contact with the "patient" object (colored in yellow) as the scene unfolds. The classification accuracy of the trained readout model on the Testing set scene representations is reported. For detailed descriptions of the VQA evaluation, refer to their paper [3].

D Implementation Details

D.1 Baselines

We build our model on SlotFormer [53] and SlotDiffusion [54] for CLEVRER and Physion dataset, respectively. Their implementations are available online.[1][2]

Stochastic SAVi (StoSAVi). As described by Wu et al. [53], vanilla SAVi [33] occasionally fails to capture objects newly entering the scene. Wu et al. [53] explains that this is caused by the more than one "empty" slots competing against each other to attend to the newly entered object, resulting in multiple slots representing the same object. To solve this problem, Wu et al. [53] proposes a stochastic version of SAVi, in which slots are initialized conditioned on previous timesteps added with a sampling procedure.

Specifically, the output of the prior network is processed through a two-layer MLP with Layer Normalization [2] to predict the mean and log variance of the initial slots at the next timestep:

$$\tilde{s}_t^k \sim \mathcal{N}(\mu_t^k, \{\log \sigma_t^2\}^k)\ ,\ (\mu_t^k, \{\log \sigma_t^2\}^k) = \text{MLP}(f_{\text{prior}}(s_{t-1}^k)) \tag{6}$$

where f_{prior} is some network used to condition slots on previous timesteps.

The model is optimized by adding a KL divergence loss on the predicted distribution to the image reconstruction loss. The loss only penalizes the log variance with a prior value $\hat{\sigma}$:

$$\mathcal{L}_{\text{KL}} = \frac{1}{TK} \sum_{t=1}^{T} \sum_{k=1}^{K} D_{\text{KL}}(\mathcal{N}(\mu_t^k, \{\log \sigma_t^2\}^k) \| \mathcal{N}(\mu_t^k, \{\log \hat{\sigma}^2\}^k)) \tag{7}$$

We set $\hat{\sigma} = 0.1$ for all datasets. The coefficient of this loss is set to 1×10^{-4}. We follow the same model architecture as implemented in [53].

[1] `https://github.com/pairlab/SlotFormer`

[2] `https://github.com/Wuziyi616/SlotDiffusion`

SlotDiffusion. The model is trained in two-stage manner, by first pretraining a VQVAE [48] to convert images to tokenized patches, and then train the encoder and Slot Attention architecture. We follow the same model architecture and training settings as [54], where the encoder is a modified ResNet18 encoder [33] and the decoder is LDM-based [42] trained to predict the noise ϵ added to the features z obtained by the pretrained VQVAE.

SlotFormer. After training an arbitrary object-centric model, the slots are extracted for all videos and saved offline. Then, SlotFormer is trained to predict slots at future timesteps, conditioned on burnin frames. The architecture and training strategy are kept unchanged from [53].

D.2 Proposed Approach: CA-SA

We implement our prior network using a GRU network. As we implement our method on top of StoSAVi, the initial slots at timestep t are sampled using the predicted mean and log variance which are computed as,

$$\tilde{s}_t^k \sim \mathcal{N}(\mu_t^k, \{\log \sigma_t^2\}^k) \ , \ (\mu_t^k, \{\log \sigma_t^2\}^k) = \text{MLP}(\text{GRU}_{\text{prior}}(s_{t-1}^k)). \tag{8}$$

We omit the hidden states h_t for simplicity. Following StoSAVi, the KL divergence loss is added to the total loss.

As described in section 3.1, the consistency loss is calculated per slot at each timestep and averaged over them. The coefficient of the loss term is set to $\lambda = 0.1$. To use image reconstruction loss when training the autoregressive transformer and to visualize the predicted slots, we train a CNN-based spatial broadcast decoder separately. This decoder is trained using reconstruction loss in image space, and the loss is not backpropagated to the encoder.

We follow Slotformer [53] approach to evaluate VP and VQA downstream tasks. Specifically, we first train CA-SA, then train the autoregressive Transformer as described in subsection A.2 using the inferred slots from the model. We validate both downstream tasks on CLEVRER [57] and Physion [3] datasets.

For CLEVRER dataset, we apply CA-SA on top of SlotFormer [53], while for Physion we use SlotDiffusion [54] as backbone model, as they are the state-of-the-art models on respective datasets. To have a fair comparison we adopt the spatial broadcast decoder used by Slotformer [53] and the conditional latent diffusion model used by SlotDiffusion [54], respectively.

To perform the VQA task, we train an auxiliary model using the slot representations generated by the autoregressive Transformer as inputs. On CLEVRER VQA task, we employ Aloe [11], a Transformer-based architecture that uses slot representations from input frames and text tokens of the question to predict the answer. For predictive questions, we use the trained Transformer to predict slots at future timesteps, and feed them to Aloe. For other questions, we follow the implementation of Aloe. On Physion VQA task, we follow the official protocol by training a readout model on generated slots, as there is no language involved in the task. Following [53], we implement a readout model which consists of a MLP applied on every two slots to extract relations between slots and a max-pool operation which is invariant to input permutations.

On CLEVRER, the training of CA-SA using CNN encoder takes 8 hours to train on 4 V100 GPUs. The training of the autoregressive transformer takes approximately 2 days with the same GPU setup. The training of VQA model takes 3 hours. On Physion, the initial training of VQVAE takes 20 hours. The training of SlotDiffusion requires 30 hours of training on 8 A100 GPUs. The training of the autoregressive transformer takes approximately 15 hours on 4 V100s. The training of the readout model finishes in less than 5 minutes.

Table 7 and Table 8 describes the hyperparameters used in our the experiments.

D.3 Experimental Setup

Video Prediction Task. We compare CA-SA with three state-of-the-art, OC models, SAVi-dyn, SAVi + SlotFormer, and SlotDiffusion + SlotFormer. SAVi-dyn uses SAVi [33] as the encoder and combines with a Transformer-LSTM to generate future slots. SAVi + SlotFormer and SlotDiffusion + SlotFormer combine respective models. For CLEVRER, the stochastic verison of SAVi was used in order to accomodate to new objects entering the scene during rollout.

Table 7: Hyperparameters used to train different encoders on each dataset.

Dataset	CLEVRER	Physion
Image encoder	ResNet18	ResNet18
Image resolution (H, W)	$(64, 64)$	$(128, 128)$
Length of sequence T	6	3
# of features $H'W'$	4096	1024
Feature dimension D_{enc}	128	192
# of slots K	7	8
# of slot attention iteration M	3	2
Slot dimension D_{slot}	128	192
Batch size	64	48
Training epochs	12	10

Table 8: Hyperparameters used to train autoregressive transformer on each dataset.

Dataset	CLEVRER	Physion
Burnin frames T	6	10
Rollout frames L	15	10
Batch size	64	128
Training epochs	80	25
# of layers	4	12
# of heads	8	8
Dimension	256	256
FFN dimension	1024	1024

For CLEVRER, we use PSNR, SSIM, and LPIPS to evaluate the visual quality of the frames generated by each model, and ARI, FG-ARI, FG-mIoU, and AR for evaluation of object-level segmentation quality. For Physion, following [54], we report visual quality metrics only, MSE, LPIPS, and FVD [47].

We follow [53, 54] with the evaluation protocol for both datasets. On CLEVRER, we use 6 burn-in timesteps to condition the model and then perform a rollout to predict the next slots for 10 steps. On Physion, the model was trained using 15 burn-in and 10 rollout timesteps. The predicted slots were decoded to images using the SBD and compared with the ground truth ones.

Video Question Answering Task. For both datasets, we apply CA-SA on top of their respective state-of-the-art model. On CLEVRER, we compare against SlotFormer + Aloe (denoted as SF + Aloe) [53]. SF + Aloe first trains StoSAVi as the feature extractor, followed by SlotFormer. Then, the predicted slots from SlotFormer and text tokens of the question are used to train Aloe, a Transformer-based VQA model. For Physion, we select SlotDiffusion + SlotFormer as the baseline model (SD + SF) [54]. This model first trains SlotDiffusion as the feature extractor, followed by SlotFormer, and finally a readout model using the predicted slots.

We report two types of average accuracy on CLEVRER VQA task, per-option (per opt.) and per-question (per ques.), as the VQA task includes multiple choice questions with more than one possible answers. The per option accuracy assesses the model's overall correctness in selecting individual options across all questions. Conversely, the per question accuracy measures correctness on a question-by-question basis, necessitating the accurate selection of all answer choices for each question. For Physion VQA task, we report the accuracy when using only burn-in frames (denoted as Obs.) and using burn-in frames and rollout frames (Dyn.).

We follow the implementation of Wu et al. [53] for evaluation on both datasets. On CLEVRER, we train Aloe [11] using the predicted slots by SlotFormer, generated by the procedure described in subsection A.2. The slots are concatenated with the text tokens of the questions and then fed to Aloe. On Physion, we train a readout model which receives every two predicted slots at each timestep as

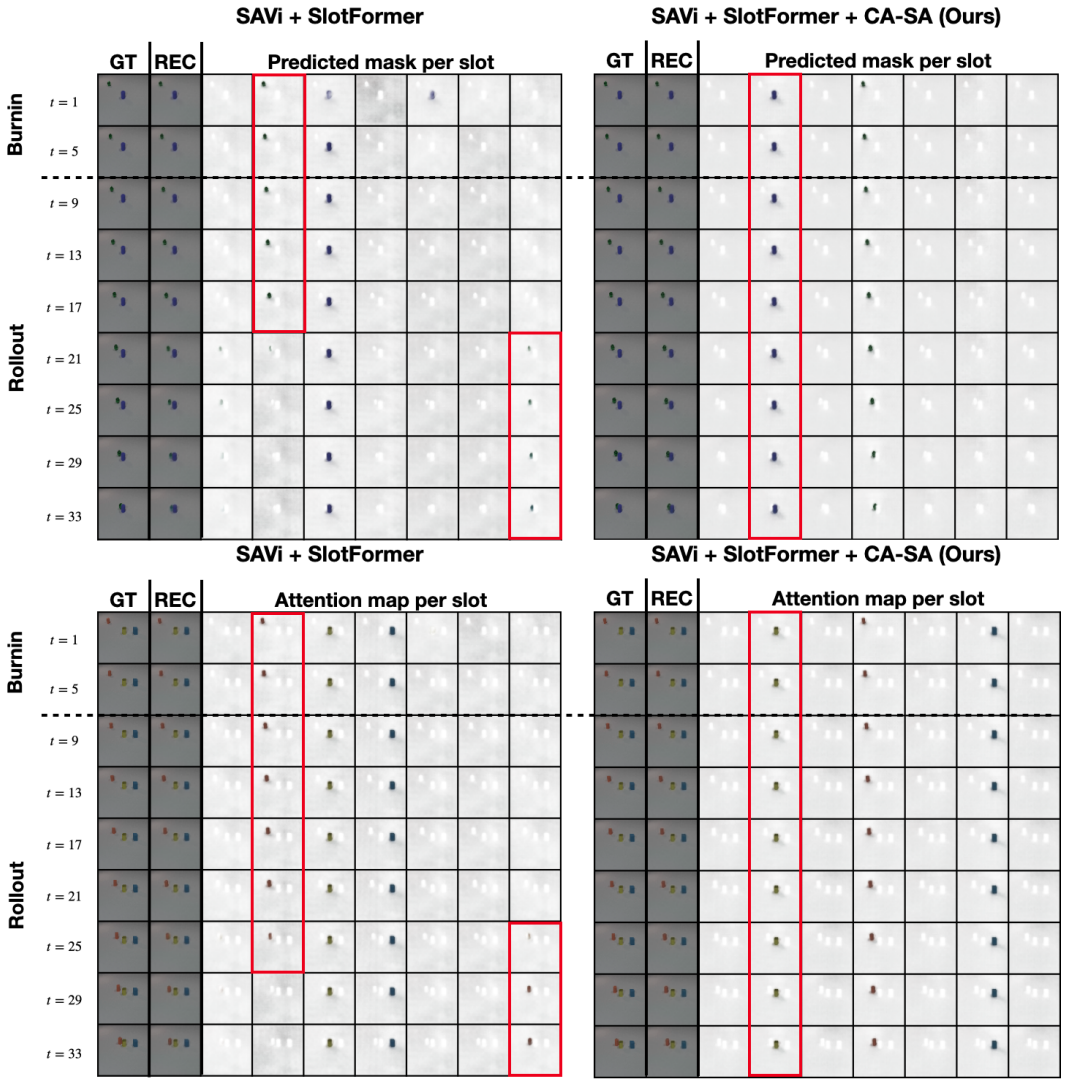

Figure 4: More generation results and predicted masks on CLEVRER. Red square indicate slots which temporal consistency is improved by adding CA-SA.

inputs. The outputs of the readout model are max-pooled over all pairs of slots and time to predict the answer.

E Rollout Visualizations

We provide further qualitative results of generated results and predicted attention maps on CLEVRER and Physion datasets in Figure 4 and Figure 5, respectively.

F Additional Results on VQA Task

We provide the accuracy per type of questions of CLEVRER in Table 9. We report the per-scenario accuracy on Physion for the model trained with rollouts in Table 10.

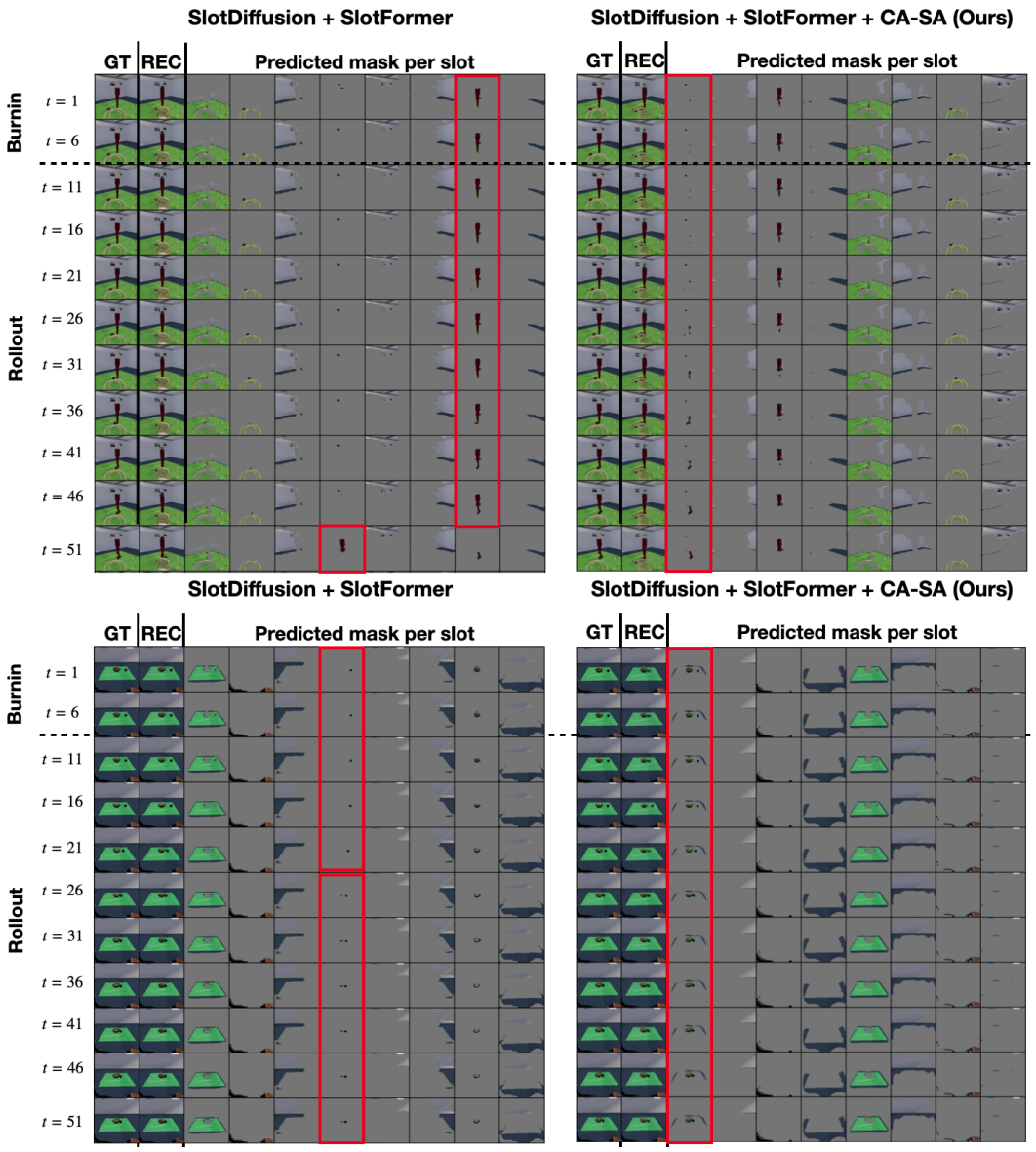

Figure 5: More generation results and predicted masks on Physion. Red square indicate slots which temporal consistency is improved by adding CA-SA.

Table 9: Detailed evaluation of video question answering task on CLEVRER dataset.

Model	Descriptive	Explanatory		Predictive		Counterfactual	
		per opt. (%)	per ques. (%)	per opt. (%)	per ques. (%)	per opt. (%)	per ques. (%)
Aloe + SlotFormer*	93.67	95.10	86.44	93.26	83.25	83.79	57.52
Aloe + SlotFormer + CA-SA(Ours)	**94.10**	**96.56**	**90.65**	**94.85**	**90.28**	**86.65**	**64.47**

Table 10: Detailed evaluation of video question answering task on Physion dataset.

Model	Collide	Contain	Dominoes	Drape	Drop	Link	Roll	Support	Avg.
SlotDiffusion + SlotFormer*	**75.3**	63.3	49.2	51.3	**65.3**	59.3	**68.0**	70.0	63.9
SlotDiffusion + SlotFormer + CA-SA(Ours)	68.7	**64.0**	**51.6**	**66.0**	60.0	**64.7**	62.7	**72.7**	**64.7**