Faster Slot Decoding using Masked Transformer

Akihiro Nakano, Masahiro Suzuki, Yutaka Matsuo
The University of Tokyo
{nakano.akihiro,masa,matsuo}@weblab.t.u-tokyo.ac.jp

Abstract

Common object-centric learning models learn a set of representations, or "slots". Recent advancements in object-centric learning have introduced autoregressive decoders to decode slots into features or images, allowing the model to learn compositional representations from more complex and realistic datasets. However, the autoregressive decoding process is time-consuming due to its sequential nature, making it difficult to apply to downstream tasks such as video generation. In this paper, we introduce MaskSDT, a masked bidirectional transformer that decodes all slots simultaneously. Our experiments on the 3D Shapes and CLEVR datasets demonstrate that our model shows improvement in reconstruction performance and generation speed, as well as comparable results in compositional generation.

1 Introduction

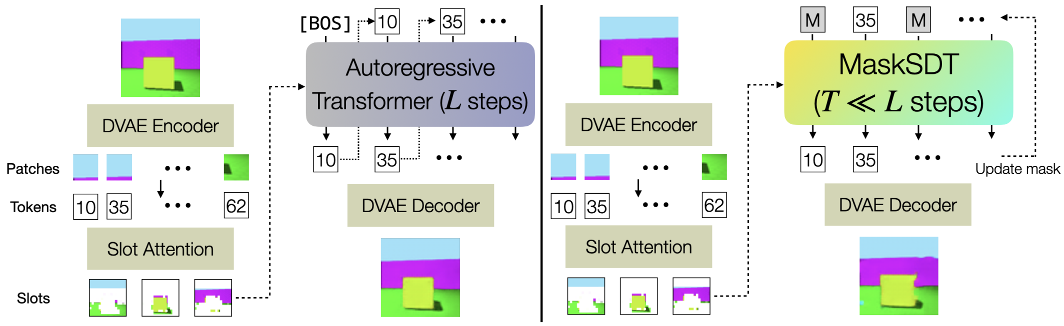
Learning compositional representations has attracted interest both within and outside the field of computer science, as it relates to how humans perceive their surroundings in terms of objects and their relationships [29, 30]. A common architecture used in object-centric learning is representing each object in an image or video as a set of representations, often referred to as "slots" [18, 2]. Due to its ability to represent the scene in a compositional manner, object-centric learning has been found useful for multiple downstream tasks, such as reasoning [20, 36, 37], planning [32, 22], and reinforcement learning [39, 7].

Slot Attention [18] is a commonly used architecture that extracts patch-level features from a CNN encoder, then applies iterative attention over features to extract slots, and decodes the slots using Spatial Broadcast Decoder [34]. In recent years, improvements for all modules have been proposed. Some works have replaced the CNN encoder with discretized encoders [10] or pretrained Vision Transformers (ViT) [31, 6] to scale to more realistic datasets [26, 28, 25, 41]. Different decoder choices have also been explored such as autoregressive transformers or diffusion models [26, 28, 37]. For example, SLATE [26] uses an autoregressive transformer that reconstructs patches from slots instead of the original image, improving object-wise disentanglement and compositional generation. Finally, several works have explored improving Slot Attention, such as optimizing the iterative attention algorithm [12] or learning quantized slot representations [27, 35].

However, using decoders other than Spatial Broadcast Decoder leads to issues with computation requirements. For example, when using an autoregressive transformer, generation requires (# of patches) steps per image. This is especially challenging in object-centric learning because the patch size is typically smaller compared to when using patches directly for downstream tasks, as the objects in the scene may vary in sizes or be partially occluded. The autoregressive property limits its suitability for downstream tasks, such as high-resolution image generation or extended video generation.

In this work, we present MaskSDT (Masked Slot Decoding Transformer), which replaces the autoregressive transformer with a bidirectional transformer. Inspired by [3], we train MaskSDT using masked token prediction. We conduct experiments using the 3D Shapes and CLEVR datasets, and show that our model improves reconstruction and generation speed. We also show that our model achieves qualitatively comparable performance on the compositional generation task.

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



(a) Generation using SLATE (b) Generation using MaskSDT

Figure 1: Generation process using SLATE (left) and MaskSDT (right). SLATE decodes L tokens one by one and take L steps per image to generate. MaskSDT generates all tokens at once with $T(\ll L)$ steps of iterative mask update.

2 Related Works

Learning to represent objects in the scene using "slots" [18] has been long explored in the literature. Several works have used discretization methods to scale to more realistic datasets [26, 28] or to improve the factorization of the learned representations [27, 35]. Using a transformer decoder has been explored both to decode tokens, used by works mentioned above, or images [23]. More recent improvements for scaling include using pretrained encoders [25, 41, 5] and using diffusion-based decoders [37, 13, 17].

Autoregressive decoding is known to suffer from the slow inference speed and sequential error accumulation, and have been extensively studied in the field of natural language processing. Non-autoregressive generation algorithms has emerged to address the challenges of autoregressive decoding, with masked token prediction recognized as a variant of this approach [4, 8, 19]. Application to images has also been explored, in which MaskGIT [3] improved largely by introducing a novel draft-and-revise algorithm [16]. MaskGIT has been applied for different tasks, such as video prediction [38], video generation [40, 33], and multimodal models [21].

3 Method

MaskSDT (Figure 1) consists of two encoder-decoder architecture in a nested structure, one to extract patch-level representations ("tokens") from images and the other to extract object-centric reprensentations ("slots") from tokens. Our model architecture mainly follows SLATE [26], while replacing the slot-to-token decoder with a transformer decoder trained with masked token prediction scheme. Motivated by its rapid generation capabilities and strong performance across a wide range of domains, we utilize the architecture and masking scheme of MaskGIT [3]. We first review the architecture of SLATE, and then introduce our model.

3.1 Preliminary: Slot-based object-centric learning using SLATE

SLATE uses Discrete VAE (DVAE) [10] to extract tokens from images. An input image $\mathbf{x}$, is processed through an encoder, f_ϕ, to calculate log probabilities, $\mathbf{o}$, for a categorical distribution with V classes. To train DVAE, a "soft" one-hot encoding $\mathbf{z}_{\text{soft}}$ is sampled from a relaxed categorical distribution [11], and decoded via a decoder, g_θ. Denoting the temperature of the relaxed categorical distribution as τ, the entire process can be written as,

$$\tilde{\mathbf{x}} = g_\theta(\mathbf{z}_{\text{soft}}) \quad \text{where} \quad \mathbf{z}_{\text{soft}} \sim \text{RelaxedCategorical}(\mathbf{o}; \tau), \quad \mathbf{o} = f_\phi(\mathbf{x}). \tag{1}$$

To compute slots, the tokens from the DVAE encoder are first mapped to embeddings, $\mathbf{e}$, using a learned dictionary. Learned positional embeddings, $\mathbf{p}_\phi$, are added to the embeddings to incorporate positional information of the tokens. Then, the embeddings are fed to Slot Attention [18] encoder to extract K slots, $\mathbf{s}_{1:K}$. This process can be written as,

$$\mathbf{s}_{1:K} = \text{SlotAttention}(\mathbf{e}) \quad \text{where} \quad \mathbf{e} = \text{Dictionary}_\phi(\mathbf{z}) + \mathbf{p}_\phi, \quad \mathbf{z} \sim \text{Categorical}(\mathbf{o}). \tag{2}$$

Then, the slots are decoded back into tokens using an autoregressive transformer [31]. Beginning with a [BOS] token, the tokens are predicted one by one, which can be expressed as,

$$\hat{\mathbf{z}}_l = \arg\max_{v\in[1,V]} \hat{\mathbf{o}}_l \quad \text{where} \quad \hat{\mathbf{o}}_l = \text{Transformer}_\theta(\hat{\mathbf{e}}_{<l}; \mathbf{s}_{1:K}), \tag{3}$$

Table 1: Evaluation of image reconstruction performance. We report MSE and FID score.

Dataset	MSE (↓)		FID (↓)	
	SLATE	MaskSDT (Ours)	SLATE	MaskSDT (Ours)
3D Shapes	9.88	**8.84**	48.67	**44.53**
CLEVR	8.85	**8.52**	51.39	**46.86**

where $\hat{\mathbf{e}}_{<l} = \text{Dictionary}_\phi(\hat{\mathbf{z}}_l) + \mathbf{p}_{\phi,l}$. The predicted tokens can then be used to generate images via the DVAE decoder, g_θ.

Overall, DVAE is trained using reconstruction loss, $\mathcal{L}_{\text{DVAE}} = \sum_{i=1}^{N}(\tilde{\mathbf{x}}_i - \mathbf{x}_i)^2$, and Slot Attention and the transformer are trained using cross-entropy loss, $\mathcal{L}_{\text{ST}} = \sum_{i=1}^{N}\sum_{l=1}^{L}\text{CE}(\mathbf{z}_{i,l}, \hat{\mathbf{o}}_{i,l})$, where L denotes the number of tokens. The entire model is trained together. Please refer to [26] for more information on training details. Index i is omitted in the equations above for brevity.

3.2 MaskSDT

Autoregressive generation is especially a bottleneck for object-centric learning, as the smaller patches are typically preferred for Slot Attention to attend to smaller objects that may be present in the scene. MaskSDT replaces the autoregressive transformer with a bidirectional transformer [4] trained on masked token prediction. Using a bidirectional transformer enables the decoder to better capture the global information between tokens. Moreover, sampling is more efficient, as multiple tokens are generated at the same time.

During training, a binary mask, $[m_l]_{l=1}^{L}$, is generated using a masking scheduler function, $\gamma(r) \in (0, 1]$. A masking ratio is first sampled, then uniformly selected $\gamma(r) \cdot L$ tokens are masked and replaced with a special `[MASK]` token. The token, $\mathbf{z}_l$, is replaced with a `[MASK]` token if $m_l = 1$, otherwise unmasked. The cross-entropy loss is replaced with a masked version, such that the loss is computed only for the masked tokens.

To generate new scenes, we start with a blank canvas with all tokens masked out and operate the following procedures iteratively for T steps; (1) Predict the log probabilities, $\hat{\mathbf{o}}_l$, for all the masked locations. (2) Sample a token based on the predicted probabilities. (3) Update masking using the mask scheduler function. (4) Obtain mask for the next iteration using the schedule from (3) and the probabilities used as “confidence” score.

In our experiments, we find that replacing the embeddings with a masked value leads to better performance compared to masking the token with a learned `[MASK]` token. We also remove the `[BOS]` token used in SLATE. For the mask scheduler function, we use the cosine function which was reported to perform best [3].

Compositional Generation. The masked transformer is conditioned on the slots extracted by Slot Attention. Therefore, following [26], we can build a visual concept library from the extracted slots, then compose concepts from the library and generate new images.

We follow the implementation of SLATE and generate new images compositionally via the following steps: (1) Collect slots from all training images. (2) Apply K-means clustering to find K concepts using cosine similarity as the distance metric. (3) To generate a new image, pick concepts from the library and randomly select a slot per concept, and decode using MaskSDT and DVAE decoder.

4 Experiments

We evaluate MaskSDT in terms of (1) image reconstruction ability, (2) computational efficiency, and (3) compositional generation ability. We compare our model with SLATE [26], which uses the same transformer-based decoder with autoregressive prediction. The evaluation is conducted on two datasets: the 3D Shapes dataset [1] and the CLEVR dataset [14]. 3D Shapes dataset consists of 400K training images of 3D objects procedurally generated from 6 ground truth independent latent factors, such as color, size, and shape. CLEVR dataset consists of 200K images of multiple objects with random colors and shapes under photorealistic lighting conditions. The images are size 64×64 and 128×128, respectively. Hyperparameters and training details are summarized in Appendix A.1.

4.1 Reconstruction

Table 1 shows the reconstruction performance and Figure 3 shows the attention maps of all models. We report MSE to evaluate how well the models preserve the contents of the original image, and

Table 2: Comparison of computation requirements using 3D Shapes dataset. All metrics were computed on a single Tesla V100 GPU using a batch size of 1.

		SLATE	MaskSDT (Ours)
Train	# of parameters	3.6M	3.7M
	Time [s]	0.891	**0.080**
Test	Time [s]	1.844	**0.286**

Table 3: Evaluation of compositional generation performance. We report FID and IS score.

Dataset	FID (↓)		IS (↑)	
	SLATE	MaskSDT (Ours)	SLATE	MaskSDT (Ours)
3D Shapes	**55.34**	115.88	3.36	**3.57**
CLEVR	**124.57**	254.75	**2.75**	2.06

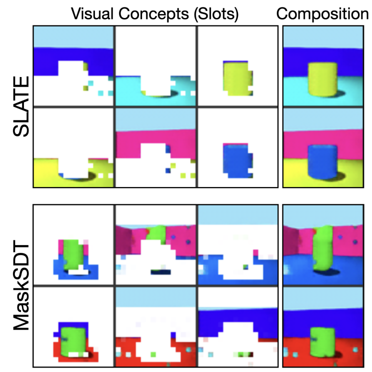


Figure 2: Comparison of compositional generation task on 3D Shapes.

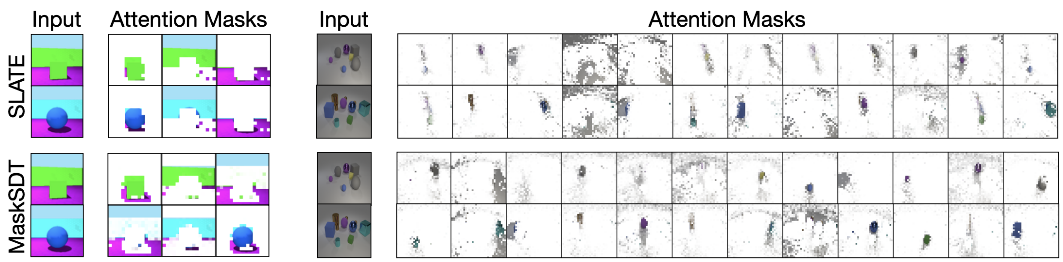


Figure 3: Visualization of attention masks for 3D Shapes dataset (left) and CLEVR dataset (right). MaskSDT produces slightly better masks for CLEVR dataset which include smaller objects and possible object occlusions.

Fréchet Inception Distance (FID) [9] to evaluate how realistic the reconstructed images are in terms of distribution distance . As the table show, our model improves both MSE and FID compared to the baseline model. MaskSDT improves especially in terms of FID as using a bidirectional transformer leads the model to capture the global context of the image while decoding. We also observe that our model improves FID score for more difficult dataset, CLEVR. We leave qualitative analysis of learned masks for future work.

4.2 Computation Efficiency

The computation requirements of the two models are summarized in Table 2. We report the number of parameters and time per training step. We also report the generation time to generate a single image. All metrics were measured on a single Tesla V100 GPU. As the table shows, our model has slightly more parameters as MaskSDT learns a separate dictionary to predict the tokens. However, our model improves training and generation speed by a large margin. This is due to our generation scheme which samples all tokens simultaneously. Although the sampling of the tokens requires $T = 5$ iterations, MaskSDT can generate scenes more efficiently.

4.3 Compositional Generation

We report the performance on compositional generation task described in section 3.2 in Table 3 and Figure 2. For this task, we report FID and Inception Score (IS) [24]. As the table shows, SLATE shows better scores except for IS on 3D Shapes dataset. However, Figure 2 shows that our model can produce realistic images in some cases. We observe that the failure case of our model is mainly wrong choice of the tokens, which may be improved by tuning hyperparameters or training setup.

5 Conclusion

In this paper, we presented MaskSDT , an object-centric model using bidirectional transformer trained on masked token prediction. We evaulated our model on three tasks, image reconstruction, computation efficiency, and compositional generation tasks, using 3D Shapes and CLEVR dataset. The results showed that while our model exceeds the baseline model for the former two tasks, optimization is needed to improve generation ability. We also leave exploring masking scheme for slots to improve out-of-domain generalization, scaling the model for more realistic datasets, and applying the model to further downstream tasks for future work.

References

[1] Chris Burgess and Hyunjik Kim. 3d shapes dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

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

[3] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, June 2019.

[5] Aniket Didolkar, Andrii Zadaianchuk, Anirudh Goyal, Mike Mozer, Yoshua Bengio, Georg Martius, and Maximilian Seitzer. Zero-shot object-centric representation learning. *arXiv preprint arXiv:2408.09162*, 2024.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[7] Stefano Ferraro, Pietro Mazzaglia, Tim Verbelen, and Bart Dhoedt. Focus: Object-centric world models for robotics manipulation. *arXiv preprint arXiv:2307.02427*, 2023.

[8] Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121, 2019.

[9] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

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

[11] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

[12] Baoxiong Jia, Yu Liu, and Siyuan Huang. Improving object-centric learning with query optimization. In *The Eleventh International Conference on Learning Representations*, 2023.

[13] Jindong Jiang, Fei Deng, Gautam Singh, and Sungjin Ahn. Object-centric slot diffusion. *arXiv preprint arXiv:2303.10834*, 2023.

[14] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross 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.

[15] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[16] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and WOOK SHIN HAN. Draft-and-revise: Effective image generation with contextual rq-transformer. *Advances in Neural Information Processing Systems*, 35:30127–30138, 2022.

[17] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pages 423–439. Springer, 2022.

[18] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in neural information processing systems*, 33:11525–11538, 2020.

[19] Elman Mansimov, Alex Wang, Sean Welleck, and Kyunghyun Cho. A generalized framework of sequence generation with application to undirected sequence models. *arXiv preprint arXiv:1905.12790*, 2019.

[20] Amin Mansouri, Jason Hartford, Yan Zhang, and Yoshua Bengio. Object centric architectures enable efficient causal representation learning. In *The Twelfth International Conference on Learning Representations*, 2024.

[21] David Mizrahi, Roman Bachmann, Oguzhan Kar, Teresa Yeo, Mingfei Gao, Afshin Dehghan, and Amir Zamir. 4m: Massively multimodal masked modeling. *Advances in Neural Information Processing Systems*, 36, 2024.

[22] Akihiro Nakano, Masahiro Suzuki, and Yutaka Matsuo. Interaction-based disentanglement of entities for object-centric world models. In *The Eleventh International Conference on Learning Representations*, 2023.

[23] Mehdi SM Sajjadi, Daniel Duckworth, Aravindh Mahendran, Sjoerd Van Steenkiste, Filip Pavetic, Mario Lucic, Leonidas J Guibas, Klaus Greff, and Thomas Kipf. Object scene representation transformer. *Advances in neural information processing systems*, 35:9512–9524, 2022.

[24] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[25] Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, et al. Bridging the gap to real-world object-centric learning. In *International Conference on Learning Representations*, 2023.

[26] Gautam Singh, Fei Deng, and Sungjin Ahn. Illiterate DALL-e learns to compose. In *International Conference on Learning Representations*, 2022.

[27] Gautam Singh, Yeongbin Kim, and Sungjin Ahn. Neural systematic binder. In *The Eleventh International Conference on Learning Representations*, 2023.

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

[29] Elizabeth S Spelke. Principles of object perception. *Cognitive science*, 14(1):29–56, 1990.

[30] Elizabeth 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.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[32] Rishi Veerapaneni, John D. Co-Reyes, Michael Chang, Michael Janner, Chelsea Finn, Jiajun Wu, Joshua Tenenbaum, and Sergey Levine. Entity abstraction in visual model-based reinforcement learning. In Leslie Pack Kaelbling, Danica Kragic, and Komei 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.

[33] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual descriptions. In *International Conference on Learning Representations*, 2023.

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

[35] Yi-Fu Wu, Minseung Lee, and Sungjin Ahn. Neural language of thought models. In *The Twelfth International Conference on Learning Representations*, 2024.

[36] Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. Slotformer: Unsupervised visual dynamics simulation with object-centric models. In *The Eleventh International Conference on Learning Representations*, 2023.

[37] Ziyi Wu, Jingyu Hu, Wuyue Lu, Igor Gilitschenski, and Animesh Garg. Slotdiffusion: Object-centric generative modeling with diffusion models. *Advances in Neural Information Processing Systems*, 36:50932–50958, 2023.

[38] Wilson Yan, Danijar Hafner, Stephen James, and Pieter Abbeel. Temporally consistent transformers for video generation, 2023.

[39] Jaesik Yoon, Yi-Fu Wu, Heechul Bae, and Sungjin Ahn. An investigation into pre-training object-centric representations for reinforcement learning. In *International Conference on Machine Learning*, pages 40147–40174. PMLR, 2023.

[40] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10459–10469, 2023.

[41] Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Object-centric learning for real-world videos by predicting temporal feature similarities. *Advances in Neural Information Processing Systems*, 36, 2023.

A Appendix / supplemental material

A.1 Hyperparameters and Training Details

The hyperparameters used for our experiments are reported in Table 4. We followed the implementation of SLATE [26] and changed only the transformer decoder architecture. Although MaskGIT [3] uses a larger transformer decoder, with 24 layers, 8 attention heads, 768 embedding dimensions and 3072 hidden dimensions, we kept our hyperparameters similar to the transformer decoder used by SLATE to measure performance fairly. The model was trained using Adam optimizer [15] with $\beta_1 = 0.9, \beta_2 = 0.999$.

We reproduced the results for the baseline model, SLATE, as only the code on 3D Shapes dataset was available. For CLEVR dataset, we also trained SLATE for the same amount of epochs.

Table 4: Hyperparameters of MaskSDT.

Dataset		3D Shapes	CLEVR
Batch Size		50	50
Epochs		20	100
Learning Rate Warmup Steps		30000	30000
Max Learning Rate		1e-4	1e-4
Gradient Clipping		1.0	1.0
Encoder	Image Size	64	128
	# of Tokens	256	1024
DVAE	Vocabulary Size	1024	4096
	Max Temperature	1.0	1.0
	Min Temperature	0.1	0.1
	Temp. Anneal Steps	30000	30000
	Learning Rate (w/o warmup)	3e-4	3e-4
Slot Attention	# of Slots	3	12
	# of Iterations	3	7
	Slot Dimension	192	192
MaskSDT	# of Layers	4	8
	# of Heads	8	8
	Embedding Dimension	192	192
	Hidden Dimension	192	192
	T (# of sampling iterations)	5	5

A.2 Ablations

As reported in MaskGIT, generation performance do not linearly increase with the number of iterations of token sampling, T. We conducted ablation using 3D Shapes dataset to investigate how FID and IS score changes with different number of iterations. As Figure 4 shows, we observed a similar trend of the score reaching a "sweet spot" then worsening again for FID. For IS score, we did not observe a similar trend. We opted to use $T = 5$ as we got reasonably low FID score with the second highest IS score. We leave further ablation of masking function and sampling iteration number for future work.

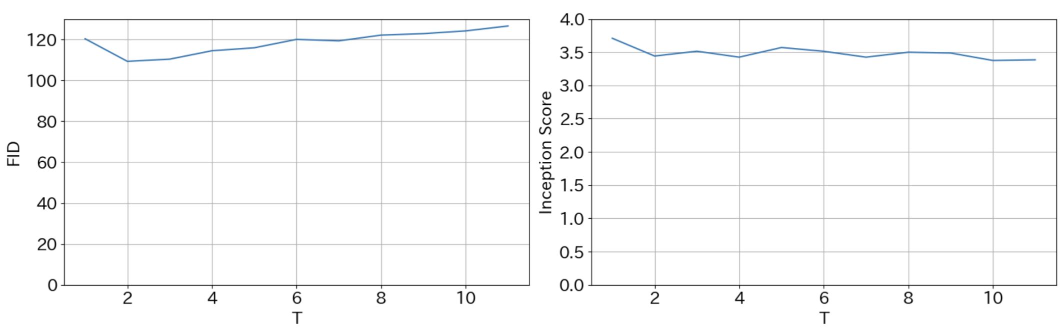


Figure 4: FID and IS score for different number of sampling iterations of MaskSDT.