How Well Can GNNs Model the World?

Christy Li
ckl@mit.edu

Gracie Sheng
grac@mit.edu

Claire Wang
clairely@mit.edu

Introduction

Effective representation learning has become increasingly crucial for developing intelligent systems that can understand and interact with complex tasks and environments. A major challenge lies in extracting meaningful and structured representations of the world from high-dimensional observations. *Structured World Models* (SWMs) are a class of models that are used to learn abstract state representations from observations in an environment. These representations offer substantial advantages over holistic, unstructured approaches because they are able to learn object-centric representations and dynamics without explicit supervision.

In our project, we revisited the GNN-based *Contrastively-trained SWM* (C-SWM) developed by Kipf et al. in [1] to investigate the architecture's scalability, generalization capabilities, and limitations in more complex scenarios. Our study presents a systematic investigation into the boundaries and failure modes of C-SWMs across three key dimensions: object-centric representation learning, physical complexity, and transfer learning. We extend the original work by testing the architecture on environments with varying numbers of objects, exploring its capacity to model increasingly complex n-body physics problems, and examining its potential for transfer learning between different Atari games.

Project Motivation

Object-centric representations have emerged as a key paradigm for understanding and modeling structured environments, especially for physical systems, where entities and their relationships are naturally separable. C-SWMs leverage object-centric representations combined with GNNs to model interactions between objects, using contrastive learning to capture underlying dynamics without requiring explicit labels [2]. This approach is particularly advantageous for environments where disentangling object-level features is critical for generalization, such as in multi-object physical simulations.

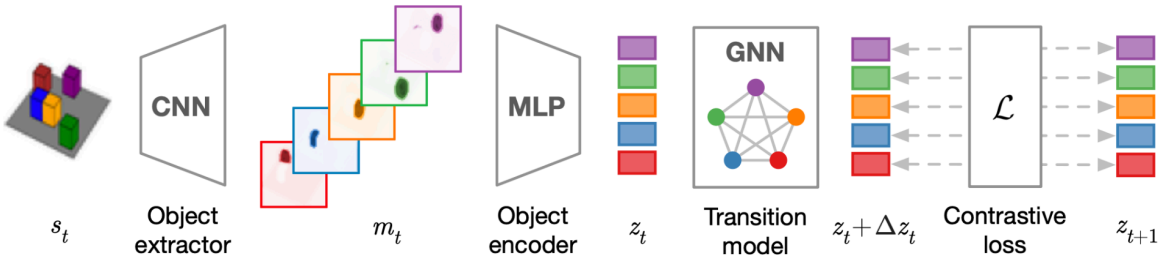


Figure 1: The C-SWM architecture consists of a CNN object extractor, an MLP object encoder, and a GNN transition model and uses contrastive loss. This example shows how the colored blocks in the 3D environment are transformed into abstract state representations by the C-SWM. Figure retrieved from original paper by Kipf et al. [1].

However, the reliance on accurate object detection and the absence of explicit mechanisms for modeling disentangled latent representations can limit the applicability of C-SWMs to complex, noisy, real-world scenarios. Addressing these limitations requires analyzing the interpretability of abstract state representations and identifying both success and failure modes of these models, which we cover in this project. Having reviewed related research summarized below, we found several endeavors which sought to improve the performance and functionality of SWMs. These were mainly "black box" experiments that compared metrics but did not explain the reasoning behind the model's behavior. In this project, our goal is to analyze *why.*

Background and Related Work

In this section, we provide context for the post-encoding portion of the C-SWM architecture, which is the central focus of our experiments.

GNNs

Graph Neural Networks (GNNs) are powerful for processing data with relational structure, such as social networks and molecular graphs, and constitute the abstract state transition model in C-SWMs. GNNs extend traditional neural networks by leveraging message-passing mechanisms to aggregate information from a node's neighbors, enabling them to capture both local and global relational patterns. The advent of the Graph Convolutional Network (GCN) exhibited the potential of GNNs in semi-supervised learning tasks on graph-structured data [2]. Subsequent advancements, like Graph Attention Networks (GATs) have refined the architecture to improve scalability and expressiveness [3][4].

Despite their strengths, GNNs have notable limitations. One primary drawback is their computational inefficiency on large-scale graphs, as message-passing schemes often require extensive memory and computation. To address this, sampling-based methods like GraphSAGE have been proposed to reduce resource overhead [5]. Another issue is the oversmoothing problem, where repeated message passing can cause node representations to become indistinguishable, limiting the depth of GNNs. Efforts to mitigate this include architectural modifications like residual connections and improved aggregation functions [6].

Additionally, GNNs are inherently limited by their dependence on graph connectivity, which can lead to suboptimal performance in graphs with noisy or incomplete structures. These challenges underline the need for ongoing research to enhance the scalability, expressiveness, and robustness of GNN models.

Contrastive Learning

Contrastive learning is a self-supervised technique that learns representations by contrasting similar and dissimilar samples, thereby extracting invariant features from raw observations. The contrastive training objective

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left(y_i \cdot \mathrm{D}^2 + (1 - y_i) \cdot \max(0, m - \mathrm{D})^2 \right)$$

N: Number of pairs.
y_i: Binary label for similarity
D: Distance between embeddings
m: Margin

ensures that positive examples —actual transitions from the environment— are closer to one another in representation space, while negative examples —randomly paired states— are distanced apart. This loss formulation allows the C-SWM to learn an implicit structure of the environment and focus on meaningful object state transitions. The margin-based penalty helps the model to handle noise.

Structured World Models

World models aim to learn an explicit representation of environment dynamics to improve sample efficiency, and allow agents to imagine or simulate potential future states and outcomes. World models were first officially introduced in Ha & Schmidhuber 2018 [7]. In this paper, they relied on a VAE and training large RNN models to learn scene information and make changes to that scene. Typically, world models consist of a representation learning module to encode the observations, a dynamics model to predict future states, and a policy to select the best actions. *Structured world models* leverage representation learning and relational dynamics to model complex environments by breaking them into entities and their interactions. It encodes objects as latent variables and uses mechanisms like graph neural networks to capture relationships, enabling better generalization to unseen configurations. Kipf et al. demonstrated the potential of C-SWMs in environments with clear object boundaries, highlighting their ability to infer object-centric representations in a self-supervised manner [1]. Furthermore, Collu et al. in 2023 introduced Slot Structured World Models (SSWM) to address limitations of previous approaches, particularly the inability of feedforward encoders to extract object-centric representations or disentangle multiple objects with similar appearances [8]. By combining Slot Attention-based object-centric encoders with latent graph-based dynamics models, SSWMs achieve superior performance in multi-step prediction tasks

Methods

To evaluate the efficacy of C-SWMs, we reproduced the experiments outlined in the implementation by Kipf et al. [1], utilizing their publicly available repository (https://github.com/tkipf/c-swm). The core experiments focused on assessing the ability of C-SWMs to learn object-centric representations and predict relational dynamics across various benchmarks. Prior to running new experiments, we first replicated the training pipeline, ensuring the alignment of parameters and experimental configurations with the original work. The primary datasets and environments used in the reproduction and in new experiments include multi-object interaction settings: *Atari Pong, Space Invaders, 3-Body Problem*.

To modernize and extend the original implementation, we updated dependencies to work with current Python and library versions, resolving compatibility issues. Additionally, we introduced custom scripts to facilitate visualization of learned embeddings and relational dynamics, providing more intuitive insights into model performance. These updates are hosted in our forked repository (https://github.com/ClaireBookworm/scene-gnns). Our code for new experiments including inference and transfer learning are also present in the repository.

To evaluate model performance, we employed standard metrics such as Hits @ 1 and Mean Reciprocal Rank (MRR), which quantify the accuracy of the model's object predictions and rank-based performance in relational reasoning tasks, respectively. These metrics provide a comprehensive assessment of the model's ability to generalize to unseen configurations and dynamics.

Experiments and Results

Latent Representation Analysis

Model	1 Step (Hits@1 / MRR)		5 Steps (Hits@1 / MRR)		10 Steps (Hits@1 / MRR)	
	New	Orig.	New	Orig.	New	Orig.
2D Shapes	1.00 / 1.000	1.00 / 1.000	1.00 / 1.000	1.00 / 1.000	1.00 / 1.000	0.99 / 1.000
Atari Pong	0.72 / 0.780	0.35 / 0.543	0.42 / 0.571	0.13 / 0.281	0.14 / 0.312	0.10 / 0.211
Space Invaders	0.31 / 0.4934	0.46 / 0.632	0.06 / 0.1603	0.11 / 0.285	0.02 / 0.1215	0.06 / 0.209
Three Body Physics	1.00 / 1.000	1.00 / 1.000	0.97 / 0.983	0.97 / 0.988	0.76 / 0.847	0.75 / 0.852

Table 1: Baselines of our ground state models in comparison to the original Kipf et al. 2020 paper results [1]. "New" represents our results and "Orig." represents the paper's results.

Pong (K = 3, K = 5)

The first game experiment we reproduced was Atari Pong, which is a 2D interactive multi-object game. Like in the original paper, we used a dataset consisting of 50x50x6 tensor observations for training and evaluation, and trained for 200 epochs. Pong includes three objects: a ball and two blocks. We replicated experiments for K=3 and K=5 where K refers to the number of object slots the model will allocate. In the

paper by Kipf et al., latent representations were not shown for complex game environments [1]. Hence, we visualized Pong in our experiments. See Figure 2 for the visualization of masks, embeddings, and state transitions for the Pong, K=5 experiment.

A key finding in this experiment was that the encoder was able to capture all three game objects according to the object masks. Unfortunately, the model appeared to struggle with distinguishing between certain objects —see objects 3 and 5— and therefore treated two objects (in this case the ball and a block) as a single object. Nevertheless, it confidently discarded unnecessary object slots —see objects 1, 2, 4. The distance between similar and dissimilar objects in embedding space also aligns with the input data, which further indicates that the objects are being distinguished. This same observation is reflected in the per-object abstract state transition graphs. Since the state and action spaces of Pong are expansive, the interpretability of the state transition graphs appears to suffer, though the plots do imply smooth and continuous transitions over time, which remains in accordance with the dynamics of the game.

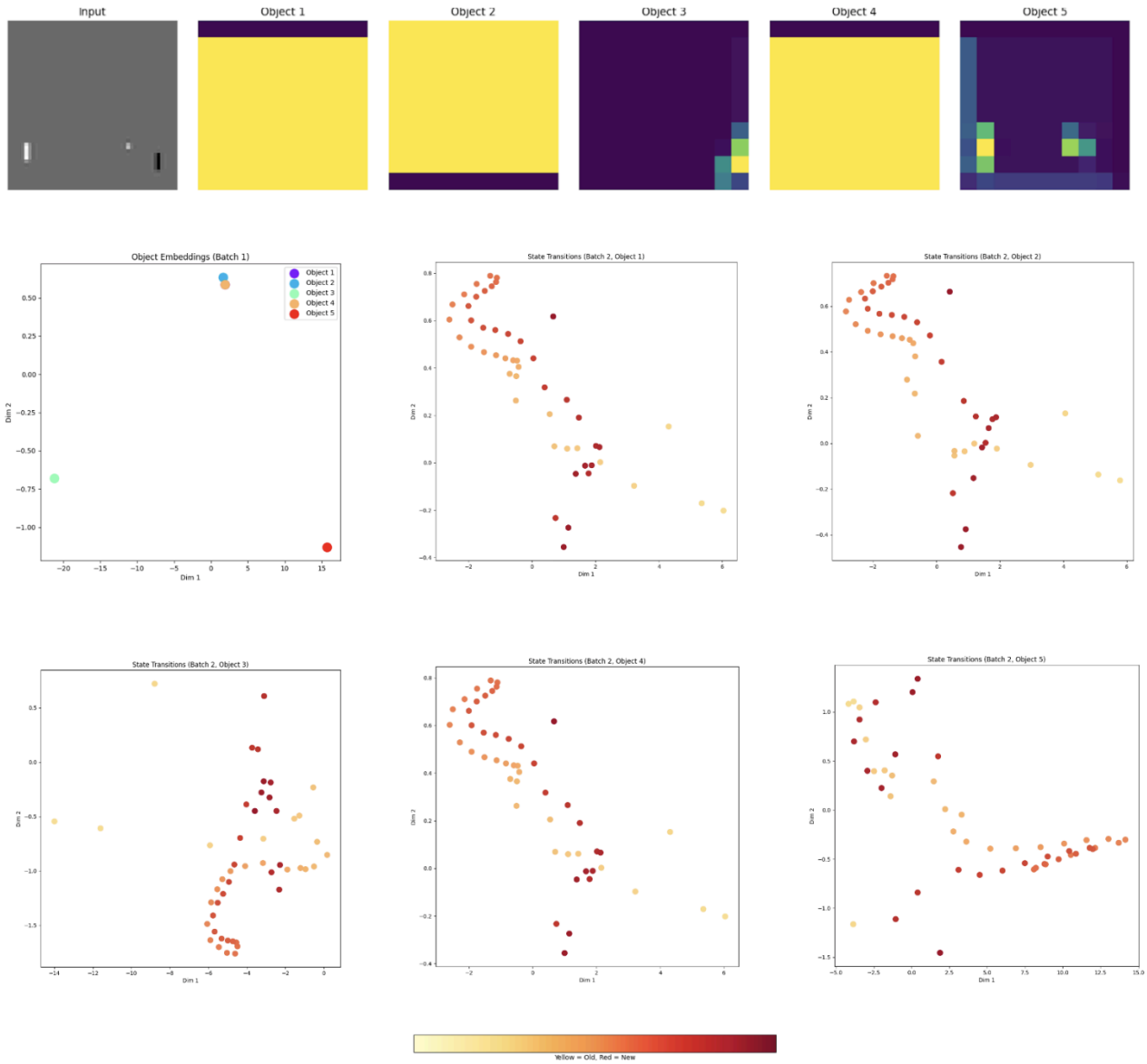


Figure 2: Atari Pong, K=5. The input, object masks, per-object abstract state transitions, and latent space object embeddings are depicted. Principal Component Analysis (PCA) was used to map latent representations onto the 2D plots.

n-Body Problem

We further investigate the C-SWM architecture's effectiveness at modeling physics scenarios. In [1], the authors report promising results for the model's ability to predict the next state of 3-body gravitational physics simulations. In contrast to the 2D-block and 3D-block grid worlds, there are no explicit actions in this environment. Instead, the model is provided with two frames of 50x50 pixel images that are consecutive in time and thus illustrate implicit action from the pair-wise gravitational forces between objects.

Our first contribution is generalizing the 3-body physics simulation environment from [9], which requires that all objects have the same mass. Our new general simulation environment is able to model *n-body* gravitational physics between objects of *variable mass*. This way, we are able to explore more diverse physics environments, including those with objects that have distinct properties.

For our experiments, we first reproduce the results from [1] with the 3-body environment and objects of equal mass. We also train C-SWM on a 3-body environment with objects of different masses to investigate if and how the model is able to represent systems in which "actions" depend on the properties of the objects (e.g. more or less massive objects in our physics simulation will follow different trajectories through space and time under the influence of the gravitational force from other objects). We hypothesize that such scenarios will be more difficult for the model to predict accurately, especially over many time steps.

Additionally, we generate data from a 2-body environment with objects of the same mass and a 2-body environment with objects of different masses. We maintain training a C-SWM model with 3 object slots on the 2-body datasets to investigate how the model compensates for more object slots than the ground truth number of relevant objects to track. We hypothesize that the model will learn some kind of "null" object whose transitions are and position has no effect on the representations of "real" objects.

Finally, we evaluate the performance of the 3-body, constant mass model on our new datasets (3-body with different masses, 2-body with the same masses, and 2-body with different masses) to investigate if the model has learned general physical laws that generalize to other physical environments. Our results are summarized in Table 2.

a)

Model	1 Step (Hits@1 / MRR)	5 Steps (Hits@1 / MRR)	10 Steps (Hits@1 / MRR)
3-body – Mass ratio 1:1:1	**1.000 / 1.000**	**0.966 / 0.983**	0.761 / 0.847
3-body – Mass ratio 1:2:3	0.999 / **1.000**	0.909 / 0.950	0.498 / 0.620
2-body – Mass ratio 1:1	**1.000 / 1.000**	0.961 / 0.978	**0.807 / 0.877**
2-body – Mass ratio 1:2	0.998 / 0.999	0.892 / 0.936	0.566 / 0.695

b)

Model	1 Step (Hits@1 / MRR)	5 Steps (Hits@1 / MRR)	10 Steps (Hits@1 / MRR)
3-body – Mass ratio 1:2:3	**0.998 / 0.999**	**0.535 / 0.713**	**0.020 / 0.082**
2-body – Mass ratio 1:1	0.830 / 0.910	0.041 / 0.120	0.007 / 0.039
2-body – Mass ratio 1:2	0.830 / 0.909	0.038 / 0.125	0.003 / 0.034

Table 2: **(a)** Ranking results from training the C-SWM model on data from a 3-body physics simulation with constant mass (mass ratio between objects is 1:1:1) and different masses (mass ratio between objects is 1:2:3) and from a 2-body physics simulation with constant mass (mass ratio between objects is 1:1) and different masses (mass ratio between objects is 1:2). **(b)** Ranking results from training the C-SWM model on data from a 3-body physics simulation with constant mass (mass ratio between objects is 1:1:1) and evaluating on each of the other physics scenarios.

We found that all models perform near perfect in predicting the correct state after 1 step, however as the number of steps increases, performance falls off across all models. We see that in the long term, the model trained on the 2-body system with constant masses performs the best followed by the 3-body system with constant masses. This may be attributed to the fact that introducing objects of variable mass creates more complicated force relationships between objects for the model to learn. In evaluating the baseline 3-body system with constant masses on the datasets of the 3-body system with different masses, the 2-body system with constant masses, and the 2-body system with different masses, we find that it performs best on the 3-body system with different masses across all time steps. In all scenarios, but especially for the 2-body systems, the model's performance falls off drastically as the number of steps increases. This seems to imply that the model is not actually learning many generalizable laws of physics as much as it is simply identifying patterns within the specific setting it was trained on.

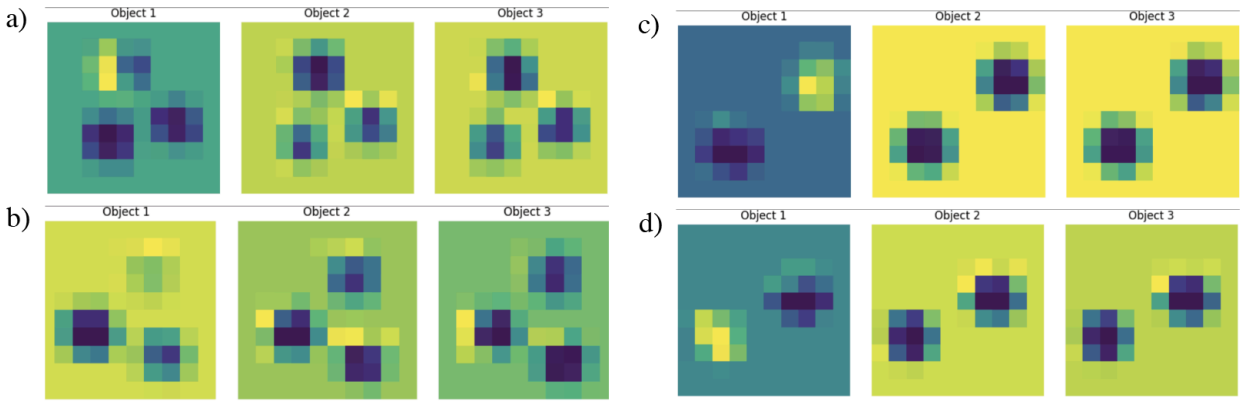


Figure 3: The object masks for each object learned by the models for **(a)** 3-body system with constant mass **(b)** 3-body system with mass ratio 1:2:3 **(c)** 2-body system with constant mass **(d)** 2-body system with mass ratio 1:2

Figure 3 shows visualizations of the object extractor learned object masks for each of the environments we tested. We see that, across all models, each "object" encoding does not actually correspond to any particular object but instead a far less interpretable representation of the image as a whole. Thus, when we trained C-SWM models with three object slots on data with only two relevant objects, we saw no major decrease in performance. Their encodings, as in the three object case, are not actually object-centric, so the model can easily compensate for these scenarios. This finding also *contradicts* a major claim in [1]: depending on the downstream task, the C-SWM architecture does not necessarily always find the object-centric embedding.

Transfer Learning

We investigated the generalization capabilities of our Contrastively-trained Structured World Model (C-SWM) across different Atari game environments. Our experiments focus on cross-training between Space Invaders and Pong to understand the model's ability to transfer learned representations across disparate visual domains. We froze the object extractor and object encoder weights and only updated the transition model (GNN) weights.

Our approach involves training the C-SWM on one game environment and fine-tuning it on another. This methodology allows us to probe the model's capacity for representation transfer and understand the underlying mechanisms of structural scene understanding.

While task-specific performance showed limited transfer, we observed two critical insights: (1) The model demonstrated rapid convergence of latent representations during fine-tuning, suggesting an adaptable understanding of scene dynamics. (2) The inability to directly transfer performance highlights the complex challenges of cross-domain generalization in structured world models.

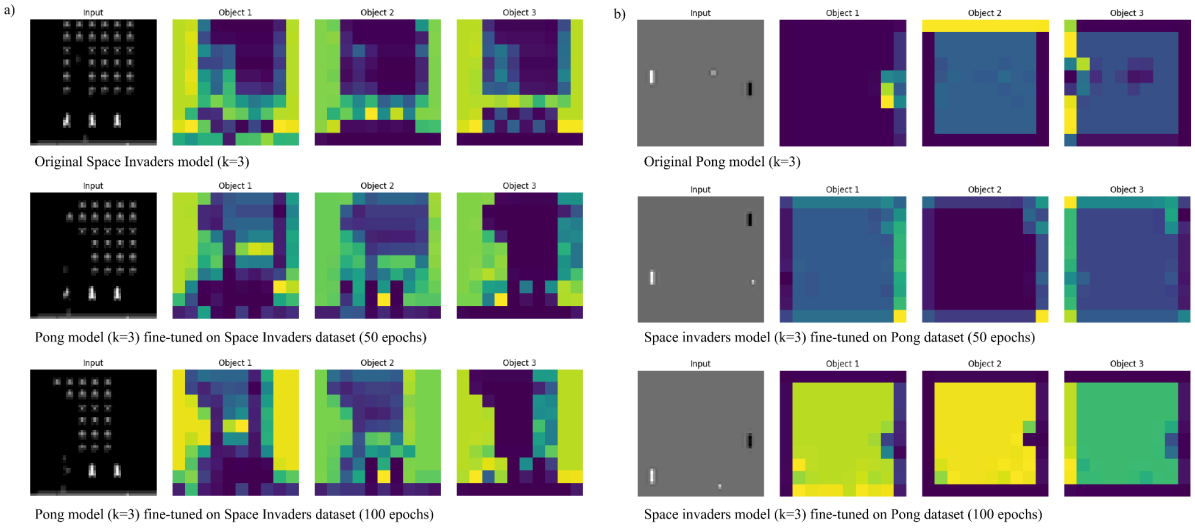
Table 3 presents a tabular summary of the model's performance, measured by the 1-step, 5-step, and 10-step "Hits@1 / MRR" metrics. This data allows for a quantitative comparison of the model's capabilities across the different training configurations and in comparison to the original Space Invaders model.

Finetuning Pong for Space Invaders	1 Step (Hits@1 / MRR)	5 Steps (Hits@1 / MRR)	10 Steps (Hits@1 / MRR)
Original Space Invaders Model	0.31 / 0.4934	0.06 / 0.1603	0.02 / 0.1215
Original Pong Model	**0.02 / 0.0579**	0.01 / 0.0519	0.01 / 0.0519
Finetuned for Space (50 epochs)	0.01 / 0.0540	0.01 / 0.0517	**0.01 / 0.0521**
Finetuned for Space (100 epochs)	0.01 / 0.0528	**0.01 / 0.0519**	0.01 / 0.0520

Table 3: Ranking results from training the C-SWM model (k=3) on the Space Invaders dataset, comparing the performance of the original model that was trained for the task, the original Atari Pong model that has not been fine-tuned at all for the task, and the Atari Pong model fine-tuned for 50 and 100 epochs.

Figure 4 provides a visual interpretation of the model's internal representations for each environment and training configuration. The model manages to learn a faithful representation of the new game environment. While the model may not achieve direct performance parity when transferred to a new task, the rapid adaptation of its latent representations suggests a possible more generalizable understanding of scene composition and relational dynamics.

Figure 4: Transfer-learned models, K=3. The input, object masks, and latent space object embeddings are depicted for the **(a)** Pong model fine-tuned on the Space Invaders dataset and **(b)** Space Invaders model fine-tuned on the Pong dataset for 50 and 100 epochs.



Discussion and Conclusion

From our experiments, we understand that C-SWMs perform well when applied to scenarios involving visually distinct objects and a fixed number of objects, where the model can effectively learn object relationships and representations. On the other hand, performance may degrade in more complex environments, such as games, where objects can vary greatly in appearance or number. This limitation arises from the model's reliance on a predefined number of objects, which is fixed during training. In terms of the parameter K, which represents the number of objects or "nodes" in the GNN, it plays a crucial role in the model's design. As seen in the Pong visualizations, the model learns to discard objects that do not exist in a given scene, effectively making use of only the relevant slots. Thus, the value of K dictates the capacity of the model to handle different numbers of objects, and it may be the case that not all slots are necessary for certain tasks, which could lead to inefficiencies. Future work could explore how to dynamically adjust K or implement more flexible representations to better accommodate variable object numbers in more complex scenes like games.

The n-body experiments demonstrated the limitations of the C-SWM architecture in modeling increasingly complex physical systems. We see performance drop off significantly as the model tries to predict more time steps into the future or additional complications such as variable masses are added. Further, we found that the downstream task has a large effect on the interpretability of C-SWM embeddings. Although the 3-body and 2-body physics simulations involved only a small number of visually distinct objects, the encodings learned by the model were no longer object-centric.

While the fine-tuned models for Pong and Space Invaders did not perform better, more experiments must be done before we can make conclusions on the effects of transfer learning with C-SWMs. The GNN component of the C-SWM is likely a bottleneck for generalizing just between two different games since there is such a drastic change in the latent representations of the objects as well as game mechanics. Future work would be to compare the models' performances on differing K values (e.g., K=1, K=5) and train it for more epochs. In addition, we fine-tuned the Atari Pong model on just 100 episodes of the Space Invaders dataset because of compute limits, but we could train it on the full 1000 episodes in the future.

Future Work

Future research on C-SWMs could explore the integration of hard negative mining and the InfoNCE loss function to further improve contrastive learning. By focusing on more challenging negative examples during training by selecting multiple random examples and choosing the one that is the most distant from our current state, the model could learn to differentiate between similar objects more effectively. We had tested the difference in performance between InfoNCE in the Pong (K=3) model and saw only improvements when evaluating the next 10 steps and decreases in accuracy for 1 and 5 steps. However, we did not train any further models, which means this could be a good starting point for future work.

Additionally, as noted in previous literature, it was observed that the model struggled with discerning *visually* similar objects, which could lead to confusion propagating to downstream tasks. To address this limitation, we propose investigating alternative encoder architectures, such as attention-based models or

more advanced convolutional neural networks, that could better capture fine-grained visual features and improve the model's ability to distinguish between subtle differences in object appearance. These improvements could enhance the generalization capabilities of C-SWMs and make them more robust to a wider range of visual challenges, particularly on more complex datasets.

Works Cited

[1] Kipf, T., van der Pol, E., & Welling, M. (2020). Contrastive Learning of Structured World Models. arXiv preprint arXiv:1911.12247.
[2] Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. International Conference on Learning Representations (ICLR).
[3] Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2018). Graph attention networks. International Conference on Learning Representations (ICLR).
[4] Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2019). How powerful are graph neural networks? International Conference on Learning Representations (ICLR).
[5] Hamilton, W. L., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. Advances in Neural Information Processing Systems (NeurIPS).
[6] Li, Q., Han, Z., & Wu, X.-M. (2018). Deeper insights into graph convolutional networks for semi-supervised learning. AAAI Conference on Artificial Intelligence.
[7] Ha, D., & Schmidhuber, J. (2018). World models. arXiv preprint arXiv:1803.10122.
[8] Collu, C., Caselles-Dupré, J., & Houthooft, R. (2020). Slot-structured world models. *OpenReview*.
[9] Miguel Jaques, Michael Burke, and Timothy Hospedales. Physics-as-inverse-graphics: Joint unsupervised learning of objects and physics from video. arXiv preprint arXiv:1905.11169, 2019.