Part I

Appendix

A Dataset Documentation

We open-source our environment, data, code and instructions on how to run them [1]. We do not provide the data as a downloadable file, it can be generated using the instructions in the repository. We provide instructions to reproduce the results of our benchmark experiments. The data is provided under the MIT license. We bear all responsibility in-case the dataset leads to any violation of rights. The metadata for the dataset can also be found in the given github repository.

Intended Use. The intended use of this dataset is for causal learning research in model-based RL. We hope that this dataset can help to speed up discovery of novel methods that can learn causal relations in RL environments.

Reading the Data. The data is generated and stored in HDF5 format. [2], it can be accessed using the h5py python package [3]. We provide the code for reading the data.

B A short review to Structured Causal Models

Causal modeling. A Structural Causal Model (SCM) [Peters et al., 2017] over a finite number M of random variables X_i is a function that maps from the jointly-independent noise N_i and parents (direct causes) $X_{pa(i,C)}$ of X_i to X_i. The matrix $C \in \{0,1\}^{M \times M}$ represents the adjacency matrix (structure) of the graph, such that $c_{ij} = 1$ if node i has node j as a parent (equivalently, $X_j \in X_{pa(i,C)}$; i.e. X_j is a direct cause of X_i).

$$X_i := f_i(X_{pa(i,C)}, N_i) \ , \quad \forall i \in \{0, \dots, M-1\} \tag{1}$$

Causal structure discovery is the recovery of ground-truth C from observational and/or interventional studies.

Interventions. An intervention on a variable X_i changes the function f_i that maps from the causal parents of X_i and the independent noise $((X_{pa(i,C)}, N_i))$ to X_i. There are several common types of interventions available [Eaton and Murphy, 2007]: *No intervention:* only observational data is obtained from the ground truth model. *Perfect:* the value of a single or several variables is fixed and then ancestral sampling is performed on the other variables. *Imperfect:* the conditional distribution of the variable on which the intervention is performed is changed. All our experiments are performed with perfect interventions (aka. setting the state of a variable to a particular value, for example location or color), as they are the most common type of interventions in RL.

C Ranking based Evaluation

Apart from standard reconstruction loss, we also provide ranking results based on the evaluation metrics followed by Kipf et al. [2019]. Given observations at two different time steps, these metrics capture how close is the predicted transition in the embedding space to the embedding of the true observation obtained through the true environment transitions. Here the notion of closeness is defined as ranking from a large buffer of states under euclidean norm.

C.1 Hits at Rank 1 (H@1)

This score is 1 for a particular example if the predicted state representation is nearest to the encoded true observation and 0 otherwise. Thus, it measures whether the rank of the predicted representation is equal to 1 or not, where ranking is done over all reference state representations by distance to the true state representation. We report the average of this score over the test set.

[1] https://github.com/dido1998/CausalMBRL

[2] https://www.hdfgroup.org/solutions/hdf5/

[3] https://pypi.org/project/h5py/

C.2 Mean Reciprocal Rank (MRR)

This is defined as the average inverse rank, i.e, MRR $= \frac{1}{N}\sum_{n=1}^{N}\frac{1}{\text{rank}_n}$ where rank_n is the rank of the n^{th} sample of the test set where ranking is done over all reference state representations.

D Model-free RL algorithms

We evaluated model-free reinforcement learning models on our tasks. The reasons are two-folds. First, this would allow us to gain insights on the difficulty of the tasks (aka, whether a model-free agent without an explicit world-model could solve the task. Another one is that this would allow us to compare the representation learned by various models and check if they would help to improve performances for model-free RL algorithms.

D.1 Methodology

We evaluated the popular Proximal Policy Optimization (PPO)[Schulman et al., 2017] on our tasks under 3 different settings. We first trained the PPO algorithm from scratch, this let us check how the difficult our tasks are for model-free algorithms, it also acts as a good baseline for comparing representations learned by a pretrained encoder verus representations learned using a pure RL objective. In the second setting, the encoder that maps from input pixels to hidden state of the LSTM in the PPO algorithms is replaced by the encoder of a pretrained model described in Section 3. These pretrained models (Autoencoders, VAE, modular networks and GNNs) are trained to predict next-step observations. We then freeze the parameters of these encoders while training the entire PPO model and we evaluate the model on the RL performance. The last setting takes in a pretrained encoder similar to the previous setting, except that it allows gradients to be passed back into the encoder during training. The last 2 settings allow us to compare whether the representation learned by various models (Autoencoders, VAE, Modular networks and GNNs) are helpful for downstream RL tasks.

E Reward Prediction Evaluation

Below, we provide the methodology of training the reward predictor and doing evaluation based on it as well as further implementation details relevant to our particular set of environments.

E.1 Methodology

For downstream RL evaluation, we consider learning a reward predictor and then performing planning based on taking greedy actions in the direction of immediate highest reward (inspired from Watters et al. [2019]). For our tasks, the reward is a function of the next state and the target state but not the action. For example, in physics environment the reward is the average distance between the objects in their current configuration and a target configuration. Similarly, for chemistry environment it is the number of color matches between the current state and the target state.

More concretely, we learn a reward predictor function (parameterized by a single layered MLP) that takes as input the current state as well as the target state of the world and tries to predict the reward for the current state. This reward predictor is learned in a supervised way and all the other weights (encoder, decoder, transition models) are kept fixed during this training. Thus, it is only possible to learn a good reward predictor if the encoder model captures the important aspects of the objects from the raw image.

Given the current encoded state of the world, we consider all possible actions and transitions according to them in the latent space (using the learned transition model). After the transition, we use the learned reward predictor to predict the reward for the (new state, target state) pair. This gives us the immediate reward obtained from each action. Having obtained those rewards, our policy is to just greedily take the action that gives us the best immediate reward. Note that in our reward setting (dense and/or partial rewards) this is typically a good policy as can be seen in the oracle (greedy) performance (where we take actions according to the true reward).

For training, we consider the supervised L_1 loss optimized using the Adam Optimizer -

$$\mathcal{L}_{Reward\ Predictor}(\theta) = \|f_\theta(s_t, s_{target}) - r(x_t, x_{target})\|_1$$
$$s_t = \text{Encoder}(x_t)$$
$$s_{target} = \text{Encoder}(x_{target})$$

where $r(\cdot, \cdot)$ is the true reward function.

For evaluation, we consider the true final reward as well as the success rate obtained under policy π where π is implicitly defined using the learned reward function f_θ as follows -

$$\pi(s_t, s_{target}) = \arg\max_{a \in \mathcal{A}} f_\theta(\text{Transition}(s_t, a), s_{target})$$

We leave the formulation of training a value function estimator using a TD-learning objective as an important future work.

E.2 Implementation Details

For all the environments, when training a reward predictor we consider a starting state of the environment and the state of the environment obtained after doing 10 random actions. Given the starting state and the target state, we use the dense reward obtained in the configuration to act as the supervision signal for training of the reward predictor model.

For physics environment, we consider the reward to be the average distance of objects from their target configurations. Whereas, for the chemistry environment we consider the number of partial matches between the two states as the reward function.

For evaluation on downstream RL tasks, for k^{th} step prediction, we consider targets that are generated from k random actions in the environment. We also report baseline performances of a random policy as well as an optimal policy. For the physics environment, we set the optimal policy to be the one step greedy policy based on the true reward while for the chemistry environment, we consider the same actions that led to the target configuration to be the optimal policy. Note that since the chemistry environment is stochastic, the same actions may not lead to the same state. Hence any loss in performance even after performing optimal actions is due to the data uncertainty that arises due to the stochasticity.

F Model setups and training procedure

F.1 Model Based Experiments

For our model based experiments, we consider four models that encode different inductive biases -

- Autoencoders (AE) - Monolithic model that compresses everything into a single entity.
- Variational Autoencoders (VAE) - Similar to Autoencoders but with regularization to stay close to a prior distribution in latent space.
- Modular Model (Modular) - Has a separate representation for each object and can be used to capture interactions between multiple sets of objects.
- Graph Neural Networks (GNN) - Also has an object-wise representation but can capture only pairwise interactions between objects.

Each model has an encoder-decoder model as well as a transition model. The encoder-decoder model is aimed at inferring the high level causal variables from raw pixel data whereas the transition model is tasked with controlling how the encoded state transitions based on the actions taken. We build all our models on the architectural backbone provided by Kipf et al. [2019].

The encoder model is a convolutional neural network followed by a 3-layered MLP (Table 1). It outputs a single representation in case of monolithic models and an object-wise representation (i.e. separate for each object) in case of modular networks and graph neural networks.

Type	channels	activation	stride
Conv2D 9×9	512	Leaky Relu	1
BatchNorm2D	-	-	-
Conv2D 5×5	M(number of objects)	Sigmoid	5

Table 1: Architecture of the encoder used for the world models.

Type	channels	activation	stride
Linear	512	Relu	-
Linear	512	Relu	-
Linear	$M \times 10 \times 10$	-	-
ConvTranspose2D 5×5	512	Relu	5
BatchNorm2D	-	-	-
ConvTranspose2D 9×9	50	-	1

Table 2: Architecture of the decoder used for the world models.

The decoder model (if used - refer Appendix F.2) takes either a single representation (in case of monolithic models) or object-wise representations (in case of modular networks / GNNs) and outputs an image as close as possible to the input image. The structure of the decoder is detailed in Table 2.

We follow the *medium* encoder-decoder structure followed by Kipf et al. [2019]. For embedding dimension, we use a fixed embedding dimension of 32 per object where the number of objects are specified by the environment description. For example, if we have 3 objects in the environment, then the embedding dimension of Autoencoder based models is 96 while it is 32 per object for Modular/GNN models.

Mathematically, given an observation x_t, the encoder maps the observation to its latent representation s_t which is either monolithic or modular. Further, the decoder (if used) maps the latent representation back to the input space.

$$s_t = \text{Encoder}(x_t)$$
$$\hat{x}_t = \text{Decoder}(s_t)$$

Each architecture also has a transition model to model how a particular action affects the state of the world. Based on the current state of the world and an action taken, the transition model predicts the next state of the world. For monolithic models (AE and VAE), the transition model is a 3-layered MLP. For GNN, it is a graph neural network with only one node-to-edge and one edge-to-node information propagation, that is, it encodes only pairwise interactions. For modular models, it is a separate MLP for each object, that allows it to encode higher order interactions between multiple objects.

Mathematically, the transition (prediction of next state) from a given state s_t based on an action a_t can be shown as -

$$\hat{s}_{t+1} = \text{Transition}(s_t, a_t)$$

F.2 Training Details

We consider two methods of training for all our baseline models -

- Negative Log Likelihood *(NLL)*
- Contrastive Loss *(Decoder Free)*

For the models trained using NLL, we perform training in 3 stages. First, we do *pretraining* where only the encoder and decoder are trained to reconstruct the given image. Second, we learn the *transition* where the encoder and decoder are fixed and the transition function is trained to optimally predict the next state given the current state and action. Finally, we do *finetuning* where we train both the encoder-decoder model as well as the transition model on combined objectives of reconstructing the current images, reconstructing the images in next step as well as doing correct transitions in the latent space.

For the reconstructions, we use the binary cross entropy loss (BCE loss) while for the transitions, we use the mean squared error loss (MSE loss).

Mathematically, given the current observation x_t, the action taken a_t and the next observation obtained x_{t+1}, we first encode both the observations into the latent space as -

$$s_t = \text{Encoder}(x_t)$$
$$s_{t+1} = \text{Encoder}(x_{t+1})$$

We then perform a transition from the current step using the transition model as well as use the decoder to perform reconstructions based on the current encoded state as well as the predicted state -

$$\hat{s}_{t+1} = \text{Transition}(s_t, a_t)$$
$$\hat{x}_t = \text{Decoder}(s_t)$$
$$\hat{x}_{t+1} = \text{Decoder}(\hat{s}_{t+1})$$

Given these variables, the *pretraining*, *transition training* and the *finetuning* can be characterized as -

$$\textit{Pretraining :} \quad \underset{\text{Encoder,Decoder}}{\arg\min} \ \text{BCE}(x_t, \hat{x}_t)$$
$$\textit{Transition :} \quad \underset{\text{Transition}}{\arg\min} \ \text{MSE}(s_{t+1}, \hat{s}_{t+1})$$
$$\textit{Finetuning :} \quad \underset{\text{Encoder, Decoder, Transition}}{\arg\min} \ \text{BCE}(x_t, \hat{x}_t) + \text{MSE}(s_{t+1}, \hat{s}_{t+1}) + \text{BCE}(x_{t+1}, \hat{x}_{t+1})$$

For models trained with contrastive loss, we follow the same setup as in Kipf et al. [2019]. In this setup we don't use a decoder and instead learn everything in encoded state end-to-end. Mathematically, this can be described as the following -

$$\textit{Contrastive Training} : \quad \underset{\text{Encoder, Transition}}{\arg\min} \ H + \max(0, \gamma - \tilde{H})$$
$$H = \text{MSE}(\hat{s}_{t+1}, s_{t+1})$$
$$\tilde{H} = \text{MSE}(\tilde{s}_{t+1}, s_{t+1})$$
$$\tilde{s}_{t+1} : \text{ Negative state obtained from random shuffling of batch}$$

We train each stage for 100 epochs using Adam optimizer [Kingma and Ba, 2014] with a learning rate of 5e-4 and batch size 512.

G Physics Environment

G.1 Detailed setups

We provide an environment which consists of objects of different shapes and potentially different colors. Each object has a unique weight associated with it and only heavier objects can push lighter ones. This induces an acyclic tournament causal graph with sparse two-way interactions between the objects, which form the nodes of the graph.

More precisely, the physics environment with M objects (eg. 3) and colormap C (eg. blues) can be considered as the set $\{o_i = \{s_i, w_i, c_i, p_i\} \mid i = 1 \text{ to } M\}$ where o_i denotes the i^{th} object which is characterized by its position p_i, its shape s_i, its color c_i and its weight w_i. An edge exists from o_i to o_j if and only if $w_i > w_j$. We consider the weight of each object to be unique, thereby getting rid of cycles. The specifics of the environment are determined by how the shape, color and weight of an object are related. For our experimentation, we consider two different settings which are outlined below. However, we emphasize that the physics environment is not limited to just these specifications and can be easily extended to form more complicated relationships between the three properties.

G.2 Identity of Objects

Since we are proposing RL environments, we need to make sure that the mapping from the action space to the object space is well defined and observable / learnable. Here, we briefly discuss that it is the case in the settings of the physics environment proposed in this paper. We also discuss that in the

Unobserved environment this mapping can be very hard to learn and for this reason, we proposed another variant known as *FixedUnobserved* environment.

Our mapping from action space to object space is such that given an initialization of the environment, the first action dimension always corresponds to the heaviest object. Similarly, the second to the second heaviest and so on.

Now, in the *Observed* environment case, the heaviest object is also the darkest object in the scene so it is relatively easy for a model to infer the action to object mapping once it has learned the fact that intensity of color represents the weight of the object.

On the other hand, in the *Unobserved* case, the colors of the objects are sampled without replacement from a larger set of colors. For example, consider a 3 object environment with the set of colors to be red < green < orange < yellow where the ordering defines the ordering of the weight. Then if in one initialization has the colors (red, green, yellow) then here the first action dimension corresponds to the color red. However, another initialization of the same environment can be (green orange, yellow) and then the first action dimension would correspond to the green object. Thus, for a model to learn the action to object mapping, it has to learn this global ranking of colors. We found that this was typically hard for the models to do.

To alleviate the above complexity, we consider another setting *FixedUnobserved* where we keep the shapes of the objects fixed and unique. Here, there is an additional constraint that apart from the colors following a global ordering of weights, the unique shapes also follow a global ordering of weights and hence, this creates an easily learnable mapping.

G.3 All variables are observed

In this setting, we consider all the objects to be of the same color but different shades, eg. different shades of the color blue. The weight of each object is a monotonic function of its color intensity, meaning that darker objects are heavier.

Mathematically, given a colormap C (single color; continuous in intensity of the color), $c_i \in [0, 1]$ denotes the intensity of the color C for object i (1 being darkest; 0 lightest). Moreover, the weight of that object is given by $w_i = g(c_i)$, where g is a strictly monotonic function. Thus, darker objects are given heavier weights and thus can push lighter objects.

This setting easily allows for zero shot generalization since a model that has been trained on a subset of shades of a particular color can generalize to do well across different shades of the same color. Moreover, the shape of an object here is a distractor since the dynamics of the objects are only controlled by their colors.

G.4 Some variables are unobserved

In this setting, all objects are of distinct discrete colors drawn from a discrete colormap c. Each color is associated with a unique weight and here, too, heavier objects can push lighter ones but not vice versa.

Mathematically, given a colormap C (multiple discrete options), $c_i \in C$ denotes the color for object i such that $c_i \neq c_j \; \forall i \neq j$. Moreover, the weight of that object is given by $w_i = g(h_i)$, where g is an injective function and $g : C \to \mathbb{R}$.

This setting does not allow for zero shot generalization in the colors since whenever a new color is introduced, the agent will have to perform interventions on it to infer its place in the graph. However, similar to the observed case, the shapes of the objects act as distractors since the dynamics is only controlled by the colors.

G.5 Unobserved Variables but Fixed Shapes

In this setting, all objects are of distinct discrete colors and shapes where the set of shapes is kept constant across different episodes. Here, the weight of an object can be reflected either from its shape or its color. For example, the lightest object in the episode will always be of a fixed unique shape and it will always have the lightest color (where lightest color is defined according to the order on the color in the colormap - eg. red < blue < green)

This setting does not allow for zero shot generalization in either the colors or the shapes since whenever a new color or shape is introduced, the agent will have to perform interventions on it to infer its place in the graph.

G.6 Experimental Results

We perform experiments on a wide range of settings for the underlying causal graph for the physics environment. We categorize our findings below -

- *Graph Neural Networks (GNNs)* generally don't perform well compared to *Modular models* and *Autoencoders (AEs)* on a wide variety of metrics (ranking metrics, reconstruction loss, downstream RL task) in the setting of *likelihood based loss* (refer to Figure 8 - Figure 19 and Table 3 - 14)
- Models trained with *contrastive loss* are generally better at predictions made over longer time scales in terms of ranking metrics (refer to Figure 8 - 13 and Table 3 - 8)
- Models trained with *contrastive loss* are also generally better at downstream RL tasks as compared to those trained with *likelihood based loss*. In particular there are some settings where the former were able to do almost perfect planning while the latter weren't able to do good planning in any setting (refer to Figures 16, 18 and 19 and Tables 9, 13 and 14)
- Modular models and Graph Neural Networks scale better than the monolithic counterparts when the number of objects in the causal graph increases. Further, while the ranking metrics still remain good, we see that the planning metrics suffer by a large margin (refer to Figure 8 - 19 and Table 3 - 14)
- While Autoencoder models perform decently based on ranking metrics, they generally don't perform as well on downstream RL tasks when compared to Graph Neural Networks and Modular models (refer to Figure 14 - 19 and Table 9 - 14)
- While ranking metrics on the unobserved environment are still decent (refer to Figures 10 and 11 and Tables 5 and 6), we see that in terms of downstream RL planning, none of the models do much better than a random policy (refer to Figures 16 and 17 and Figures 16 and 17)
- We see a case where models that have very good ranking metrics over long time horizons (AE with NLL Finetune; Figure 12 and Figure 18) perform much worse on downstream RL tasks than GNNs and Modular models which had lower ranking metrics (Table 13 and Figure 18).

		1 Step			5 Steps			10 Steps		
	Model	H@1	MRR	Rec.	H@1	MRR	Rec.	H@1	MRR	Rec.
NLL	AE	$97.23_{\pm 0.37}$	$98.23_{\pm 0.28}$	$0.04_{\pm 0.0}$	$72.78_{\pm 2.5}$	$77.74_{\pm 2.14}$	$0.1_{\pm 0.01}$	$40.46_{\pm 3.48}$	$47.4_{\pm 3.37}$	$0.22_{\pm 0.01}$
	GNN	$64.86_{\pm 4.43}$	$73.39_{\pm 4.08}$	$0.11_{\pm 0.01}$	$17.73_{\pm 6.15}$	$25.44_{\pm 7.73}$	$0.33_{\pm 0.05}$	$6.4_{\pm 3.51}$	$11.05_{\pm 5.17}$	$0.44_{\pm 0.06}$
	Modular	$97.13_{\pm 0.55}$	$98.22_{\pm 0.42}$	$0.04_{\pm 0.0}$	$70.7_{\pm 9.01}$	$76.46_{\pm 7.95}$	$0.13_{\pm 0.02}$	$36.66_{\pm 9.88}$	$44.25_{\pm 10.14}$	$0.26_{\pm 0.03}$
	VAE	$49.52_{\pm 1.51}$	$58.98_{\pm 1.79}$	$0.25_{\pm 0.02}$	$1.7_{\pm 0.13}$	$3.4_{\pm 0.16}$	$1.0_{\pm 0.11}$	$0.16_{\pm 0.03}$	$0.56_{\pm 0.06}$	$1.18_{\pm 0.14}$
NLL Finetuned	AE	$98.08_{\pm 0.2}$	$98.81_{\pm 0.15}$	$0.03_{\pm 0.0}$	$80.95_{\pm 2.2}$	$84.54_{\pm 1.86}$	$0.07_{\pm 0.0}$	$51.98_{\pm 4.12}$	$57.96_{\pm 3.84}$	$0.16_{\pm 0.01}$
	GNN	$74.64_{\pm 11.03}$	$78.88_{\pm 10.19}$	$0.04_{\pm 0.0}$	$32.43_{\pm 16.24}$	$39.39_{\pm 17.45}$	$0.14_{\pm 0.05}$	$8.23_{\pm 7.15}$	$12.03_{\pm 9.29}$	$0.28_{\pm 0.07}$
	Modular	$98.16_{\pm 0.49}$	$99.0_{\pm 0.33}$	$0.03_{\pm 0.0}$	$81.49_{\pm 10.07}$	$86.17_{\pm 8.66}$	$0.07_{\pm 0.02}$	$48.7_{\pm 16.19}$	$56.48_{\pm 16.41}$	$0.17_{\pm 0.04}$
	VAE	$77.61_{\pm 16.75}$	$83.27_{\pm 13.68}$	$0.04_{\pm 0.0}$	$18.96_{\pm 13.9}$	$25.5_{\pm 17.07}$	$0.29_{\pm 0.08}$	$1.3_{\pm 1.08}$	$2.87_{\pm 1.96}$	$0.51_{\pm 0.07}$
Contrastive	AE	$82.11_{\pm 2.22}$	$88.5_{\pm 1.61}$	-	$50.0_{\pm 6.43}$	$65.2_{\pm 5.04}$	-	$34.36_{\pm 8.42}$	$51.22_{\pm 8.17}$	-
	GNN	$93.86_{\pm 9.59}$	$95.99_{\pm 6.42}$	-	$78.28_{\pm 32.39}$	$82.29_{\pm 26.85}$	-	$72.06_{\pm 39.58}$	$75.46_{\pm 35.65}$	-
	Modular	$98.73_{\pm 1.04}$	$99.31_{\pm 0.58}$	-	$94.7_{\pm 4.2}$	$97.02_{\pm 2.38}$	-	$90.6_{\pm 6.87}$	$94.45_{\pm 4.08}$	-

Table 3: Hits at Rank 1 (H@1), Mean Reciprocal Rank (MRR) *(higher is better)* and Reconstruction Error *(lower is better)* for different models and training losses for 1, 5 and 10 step prediction for the Observed Physics environment setting with 3 objects.

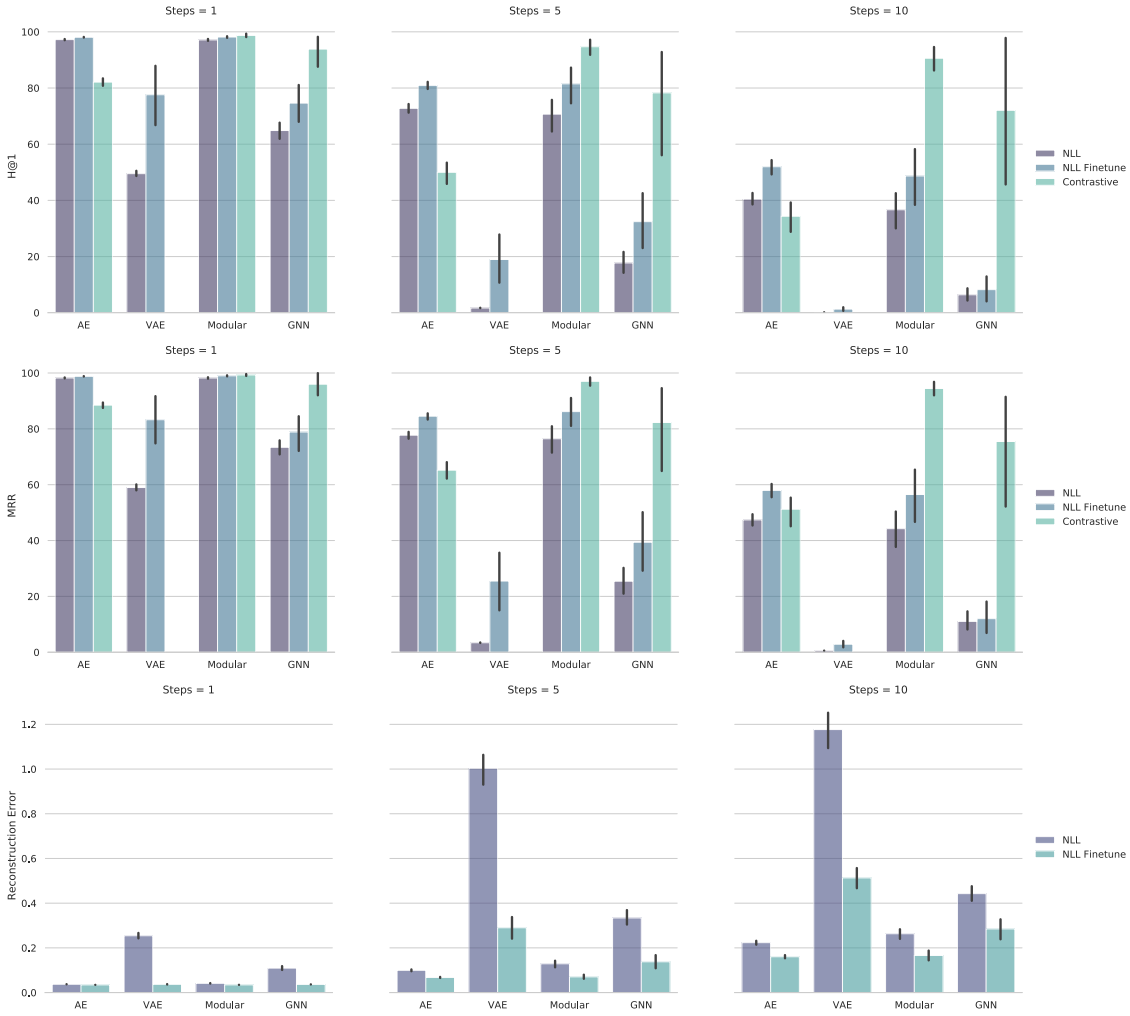

Figure 8: Hits at Rank 1 (H@1), Mean Reciprocal Rank (MRR) *(higher is better)* and Reconstruction Error *(lower is better)* for different models and training losses for 1, 5 and 10 step prediction for the Observed Physics environment setting with 3 objects.

		1 Step			5 Steps			10 Steps		
	Model	H@1	MRR	Rec.	H@1	MRR	Rec.	H@1	MRR	Rec.
NLL	AE	97.77 $\pm$1.45	98.38 $\pm$1.05	0.08 $\pm$0.01	63.88 $\pm$9.77	69.55 $\pm$9.0	0.25 $\pm$0.03	27.18 $\pm$7.09	33.6 $\pm$7.71	0.45 $\pm$0.03
	GNN	95.13 $\pm$3.02	96.95 $\pm$2.24	0.19 $\pm$0.02	41.49 $\pm$3.95	50.63 $\pm$3.93	0.47 $\pm$0.05	19.28 $\pm$2.57	26.59 $\pm$3.06	0.63 $\pm$0.07
	Modular	99.57 $\pm$0.16	99.73 $\pm$0.12	0.09 $\pm$0.0	79.14 $\pm$4.89	84.06 $\pm$4.09	0.28 $\pm$0.01	35.68 $\pm$6.99	43.82 $\pm$7.55	0.48 $\pm$0.02
	VAE	79.35 $\pm$0.48	84.38 $\pm$0.4	0.34 $\pm$0.01	6.18 $\pm$1.76	10.68 $\pm$2.25	1.62 $\pm$0.1	0.28 $\pm$0.09	0.97 $\pm$0.22	2.21 $\pm$0.13
NLL Finetuned	AE	98.29 $\pm$0.77	98.78 $\pm$0.53	0.07 $\pm$0.01	69.58 $\pm$7.23	74.59 $\pm$6.45	0.2 $\pm$0.02	31.75 $\pm$6.64	38.22 $\pm$6.97	0.39 $\pm$0.02
	GNN	97.71 $\pm$2.81	98.43 $\pm$2.13	0.07 $\pm$0.0	68.36 $\pm$18.69	73.78 $\pm$17.13	0.2 $\pm$0.05	26.52 $\pm$13.33	32.94 $\pm$14.63	0.46 $\pm$0.13
	Modular	99.65 $\pm$0.2	99.77 $\pm$0.14	0.06 $\pm$0.0	77.21 $\pm$6.81	82.08 $\pm$5.83	0.21 $\pm$0.04	23.15 $\pm$6.27	29.24 $\pm$7.12	0.53 $\pm$0.12
	VAE	68.44 $\pm$2.1	74.52 $\pm$1.6	0.09 $\pm$0.0	8.42 $\pm$1.32	12.42 $\pm$1.8	0.75 $\pm$0.03	0.58 $\pm$0.14	1.34 $\pm$0.28	1.07 $\pm$0.05
Contrastive	AE	96.12 $\pm$1.73	97.71 $\pm$1.12	-	67.36 $\pm$20.12	76.98 $\pm$15.6	-	44.65 $\pm$32.39	55.38 $\pm$29.98	-
	GNN	99.28 $\pm$0.53	99.6 $\pm$0.31	-	78.85 $\pm$7.5	84.81 $\pm$6.21	-	50.1 $\pm$9.94	60.25 $\pm$10.11	-
	Modular	99.71 $\pm$0.13	99.84 $\pm$0.08	-	84.3 $\pm$2.84	89.35 $\pm$2.26	-	52.36 $\pm$4.02	63.28 $\pm$4.28	-

Table 4: Hits at Rank 1 (H@1), Mean Reciprocal Rank (MRR) *(higher is better)* and Reconstruction Error *(lower is better)* for different models and training losses for 1, 5 and 10 step prediction for the Observed Physics environment setting with 5 objects.

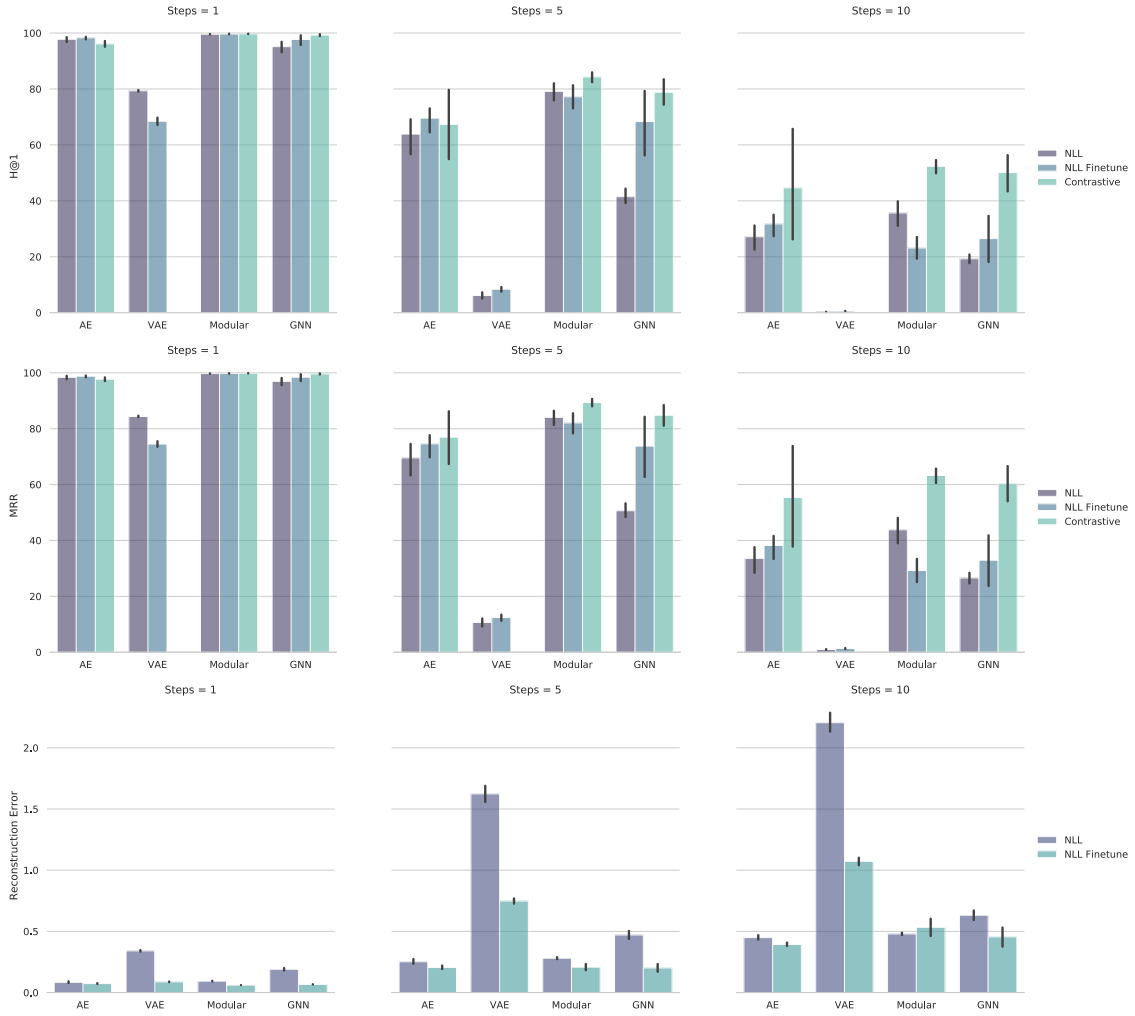

Figure 9: Hits at Rank 1 (H@1), Mean Reciprocal Rank (MRR) *(higher is better)* and Reconstruction Error *(lower is better)* for different models and training losses for 1, 5 and 10 step prediction for the Observed Physics environment setting with 5 objects.

		1 Step			5 Steps			10 Steps		
	Model	H@1	MRR	Rec.	H@1	MRR	Rec.	H@1	MRR	Rec.
NLL	AE	65.69$_{\pm 1.93}$	73.4$_{\pm 1.66}$	0.12$_{\pm 0.0}$	17.98$_{\pm 0.95}$	25.84$_{\pm 1.15}$	0.3$_{\pm 0.01}$	6.56$_{\pm 0.6}$	11.64$_{\pm 0.98}$	0.39$_{\pm 0.02}$
	GNN	62.27$_{\pm 3.7}$	70.16$_{\pm 3.5}$	0.15$_{\pm 0.01}$	19.32$_{\pm 1.64}$	26.2$_{\pm 2.14}$	0.34$_{\pm 0.02}$	8.87$_{\pm 1.35}$	14.09$_{\pm 2.03}$	0.42$_{\pm 0.02}$
	Modular	75.23$_{\pm 2.69}$	82.73$_{\pm 2.01}$	0.12$_{\pm 0.0}$	24.93$_{\pm 2.64}$	33.96$_{\pm 3.08}$	0.31$_{\pm 0.01}$	10.39$_{\pm 1.67}$	16.71$_{\pm 2.31}$	0.39$_{\pm 0.01}$
	VAE	52.83$_{\pm 1.98}$	61.68$_{\pm 1.85}$	0.28$_{\pm 0.01}$	1.96$_{\pm 0.16}$	3.92$_{\pm 0.26}$	0.88$_{\pm 0.05}$	0.19$_{\pm 0.04}$	0.62$_{\pm 0.03}$	1.0$_{\pm 0.07}$
NLL Finetuned	AE	95.35$_{\pm 1.13}$	97.02$_{\pm 0.75}$	0.06$_{\pm 0.0}$	40.92$_{\pm 7.81}$	49.77$_{\pm 7.94}$	0.21$_{\pm 0.02}$	9.41$_{\pm 4.36}$	13.92$_{\pm 5.64}$	0.35$_{\pm 0.03}$
	GNN	74.19$_{\pm 5.88}$	80.08$_{\pm 5.04}$	0.07$_{\pm 0.0}$	20.13$_{\pm 8.28}$	26.32$_{\pm 9.43}$	0.16$_{\pm 0.01}$	2.3$_{\pm 2.97}$	3.94$_{\pm 4.06}$	0.25$_{\pm 0.02}$
	Modular	94.92$_{\pm 1.84}$	96.79$_{\pm 1.24}$	0.07$_{\pm 0.0}$	27.62$_{\pm 6.53}$	34.7$_{\pm 7.51}$	0.21$_{\pm 0.02}$	2.52$_{\pm 1.21}$	4.16$_{\pm 1.74}$	0.32$_{\pm 0.03}$
	VAE	49.65$_{\pm 4.14}$	59.58$_{\pm 3.92}$	0.07$_{\pm 0.0}$	7.82$_{\pm 1.04}$	12.3$_{\pm 1.48}$	0.25$_{\pm 0.03}$	0.83$_{\pm 0.16}$	2.05$_{\pm 0.29}$	0.36$_{\pm 0.04}$
Contrastive	AE	89.77$_{\pm 3.3}$	94.0$_{\pm 2.11}$	-	37.57$_{\pm 9.15}$	53.53$_{\pm 8.72}$	-	13.87$_{\pm 7.64}$	26.54$_{\pm 10.18}$	-
	GNN	89.58$_{\pm 5.13}$	93.4$_{\pm 3.42}$	-	40.33$_{\pm 10.2}$	50.14$_{\pm 10.19}$	-	17.74$_{\pm 6.99}$	25.67$_{\pm 8.26}$	-
	Modular	96.55$_{\pm 3.09}$	97.96$_{\pm 1.96}$	-	62.15$_{\pm 12.59}$	71.49$_{\pm 11.61}$	-	31.02$_{\pm 10.94}$	42.39$_{\pm 12.6}$	-

Table 5: Hits at Rank 1 (H@1), Mean Reciprocal Rank (MRR) *(higher is better)* and Reconstruction Error *(lower is better)* for different models and training losses for 1, 5 and 10 step prediction for the Unobserved Physics environment setting with 3 objects.

Figure 10: Hits at Rank 1 (H@1), Mean Reciprocal Rank (MRR) *(higher is better)* and Reconstruction Error *(lower is better)* for different models and training losses for 1, 5 and 10 step prediction for the Unobserved Physics environment setting with 3 objects.

		1 Step			5 Steps			10 Steps		
	Model	H@1	MRR	Rec.	H@1	MRR	Rec.	H@1	MRR	Rec.
NLL	AE	89.49 $\pm$0.68	92.15 $\pm$0.62	0.15 $\pm$0.0	37.78 $\pm$1.7	45.92 $\pm$1.78	0.35 $\pm$0.01	15.04 $\pm$1.74	21.52 $\pm$2.21	0.46 $\pm$0.01
	GNN	95.76 $\pm$2.07	97.3 $\pm$1.53	0.17 $\pm$0.01	49.46 $\pm$1.98	57.92 $\pm$2.14	0.42 $\pm$0.04	28.5 $\pm$2.54	36.75 $\pm$2.99	0.54 $\pm$0.05
	Modular	98.19 $\pm$1.26	98.93 $\pm$0.81	0.15 $\pm$0.01	57.51 $\pm$5.46	66.3 $\pm$5.16	0.37 $\pm$0.03	31.67 $\pm$4.13	40.84 $\pm$4.55	0.49 $\pm$0.04
	VAE	77.21 $\pm$3.91	81.44 $\pm$3.54	0.33 $\pm$0.01	26.01 $\pm$2.63	32.41 $\pm$2.79	0.89 $\pm$0.04	9.18 $\pm$1.03	13.74 $\pm$1.25	1.18 $\pm$0.07
NLL Finetuned	AE	95.79 $\pm$0.58	97.27 $\pm$0.43	0.11 $\pm$0.0	27.77 $\pm$1.72	35.19 $\pm$1.88	0.22 $\pm$0.01	3.73 $\pm$0.45	5.92 $\pm$0.57	0.32 $\pm$0.02
	GNN	99.04 $\pm$0.72	99.43 $\pm$0.44	0.1 $\pm$0.0	58.45 $\pm$7.06	65.86 $\pm$6.56	0.2 $\pm$0.01	10.34 $\pm$3.74	15.38 $\pm$4.81	0.28 $\pm$0.01
	Modular	99.87 $\pm$0.05	99.93 $\pm$0.03	0.1 $\pm$0.0	42.15 $\pm$9.03	49.12 $\pm$9.4	0.22 $\pm$0.01	4.35 $\pm$2.47	6.32 $\pm$3.35	0.36 $\pm$0.04
	VAE	65.67 $\pm$5.74	72.42 $\pm$5.11	0.11 $\pm$0.0	15.62 $\pm$3.52	20.41 $\pm$4.25	0.3 $\pm$0.02	3.55 $\pm$1.5	5.57 $\pm$2.15	0.42 $\pm$0.03
Contrastive	AE	97.23 $\pm$0.93	98.38 $\pm$0.54	-	56.62 $\pm$5.66	68.68 $\pm$4.46	-	22.86 $\pm$5.52	35.88 $\pm$6.53	-
	GNN	99.67 $\pm$0.21	99.81 $\pm$0.12	-	82.52 $\pm$6.75	86.9 $\pm$5.33	-	55.12 $\pm$11.8	63.04 $\pm$10.74	-
	Modular	99.8 $\pm$0.14	99.89 $\pm$0.08	-	82.98 $\pm$3.25	87.44 $\pm$2.72	-	50.92 $\pm$4.73	59.51 $\pm$4.65	-

Table 6: Hits at Rank 1 (H@1), Mean Reciprocal Rank (MRR) *(higher is better)* and Reconstruction Error *(lower is better)* for different models and training losses for 1, 5 and 10 step prediction for the Unobserved Physics environment setting with 5 objects.

Figure 11: Hits at Rank 1 (H@1), Mean Reciprocal Rank (MRR) *(higher is better)* and Reconstruction Error *(lower is better)* for different models and training losses for 1, 5 and 10 step prediction for the Unobserved Physics environment setting with 5 objects.

	Model	1 Step H@1	1 Step MRR	1 Step Rec.	5 Steps H@1	5 Steps MRR	5 Steps Rec.	10 Steps H@1	10 Steps MRR	10 Steps Rec.
NLL	AE	99.0 $\pm$0.1	99.44 $\pm$0.06	0.04 $\pm$0.0	95.0 $\pm$0.41	96.81 $\pm$0.31	0.06 $\pm$0.0	84.54 $\pm$1.5	89.24 $\pm$1.2	0.1 $\pm$0.01
	GNN	70.68 $\pm$4.95	79.43 $\pm$4.13	0.11 $\pm$0.02	23.82 $\pm$7.74	33.28 $\pm$9.14	0.27 $\pm$0.05	10.11 $\pm$5.08	16.65 $\pm$6.91	0.36 $\pm$0.06
	Modular	98.03 $\pm$0.22	98.84 $\pm$0.17	0.05 $\pm$0.0	88.12 $\pm$2.65	91.8 $\pm$2.2	0.08 $\pm$0.01	68.6 $\pm$9.01	76.12 $\pm$7.96	0.12 $\pm$0.02
	VAE	53.12 $\pm$2.76	63.42 $\pm$2.58	0.21 $\pm$0.01	2.2 $\pm$0.24	4.53 $\pm$0.4	0.61 $\pm$0.05	0.2 $\pm$0.04	0.82 $\pm$0.08	0.76 $\pm$0.07
NLL Finetuned	AE	99.24 $\pm$0.08	99.59 $\pm$0.05	0.04 $\pm$0.0	96.73 $\pm$0.37	98.02 $\pm$0.22	0.05 $\pm$0.0	90.56 $\pm$1.0	93.72 $\pm$0.69	0.07 $\pm$0.0
	GNN	75.16 $\pm$12.45	79.97 $\pm$11.99	0.05 $\pm$0.0	34.78 $\pm$14.54	42.8 $\pm$16.43	0.11 $\pm$0.04	12.76 $\pm$7.38	17.88 $\pm$9.47	0.21 $\pm$0.07
	Modular	98.76 $\pm$0.15	99.35 $\pm$0.09	0.04 $\pm$0.0	91.3 $\pm$2.18	94.54 $\pm$1.63	0.06 $\pm$0.0	66.7 $\pm$7.96	75.15 $\pm$7.17	0.1 $\pm$0.01
	VAE	68.53 $\pm$13.05	76.55 $\pm$10.71	0.05 $\pm$0.0	21.38 $\pm$10.49	29.76 $\pm$12.17	0.18 $\pm$0.03	1.72 $\pm$1.0	3.85 $\pm$1.66	0.29 $\pm$0.04
Contrastive	AE	77.67 $\pm$10.51	86.21 $\pm$7.08	-	53.49 $\pm$23.05	68.11 $\pm$17.57	-	43.65 $\pm$26.31	59.13 $\pm$21.53	-
	GNN	84.94 $\pm$8.08	90.1 $\pm$5.62	-	42.88 $\pm$28.35	51.75 $\pm$24.48	-	28.06 $\pm$35.18	34.19 $\pm$32.68	-
	Modular	88.42 $\pm$6.43	93.32 $\pm$4.48	-	71.54 $\pm$17.3	83.07 $\pm$12.72	-	66.07 $\pm$20.34	79.36 $\pm$15.62	-

Table 7: Hits at Rank 1 (H@1), Mean Reciprocal Rank (MRR) *(higher is better)* and Reconstruction Error *(lower is better)* for different models and training losses for 1, 5 and 10 step prediction for the FixedUnobserved Physics environment setting with 3 objects.

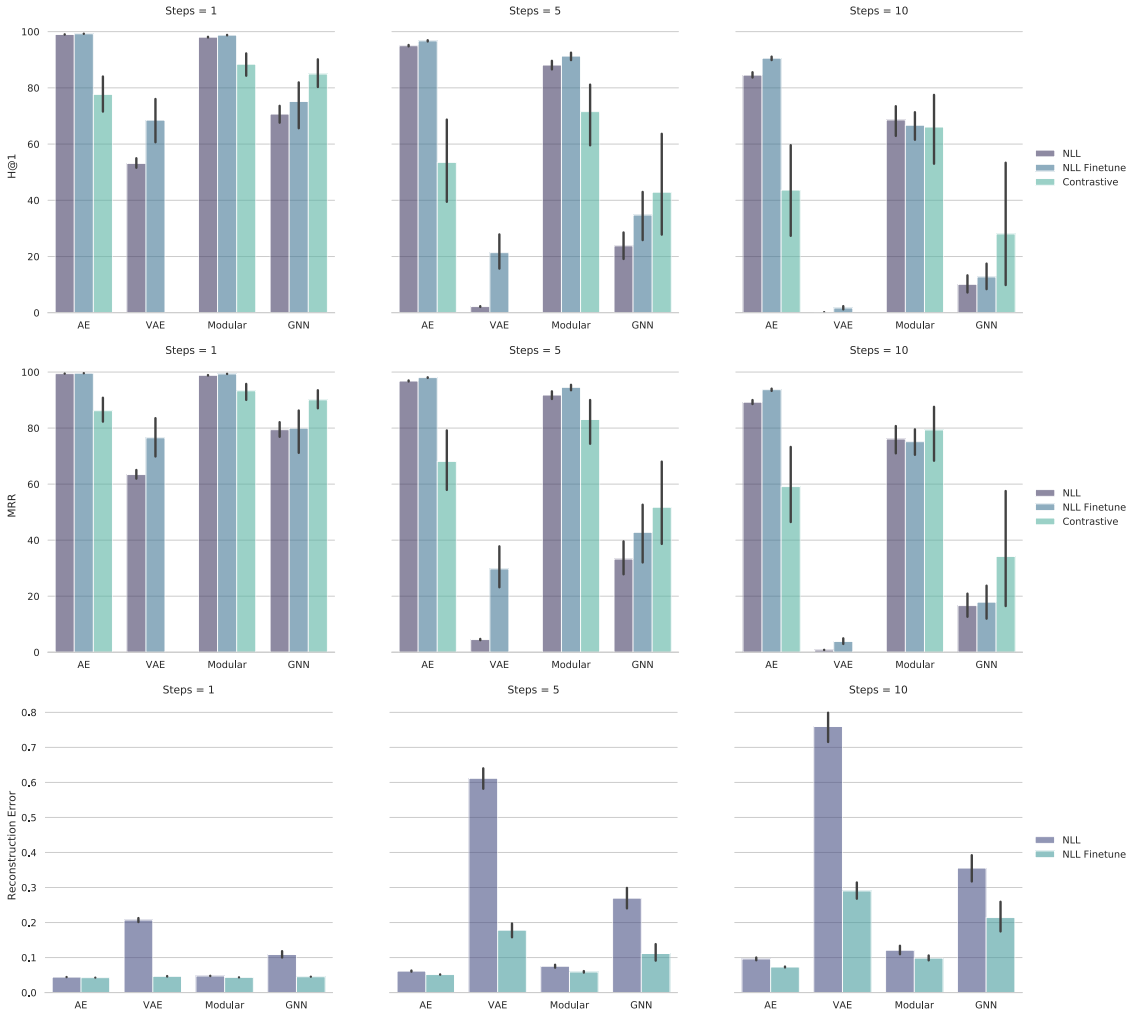

Figure 12: Hits at Rank 1 (H@1), Mean Reciprocal Rank (MRR) *(higher is better)* and Reconstruction Error *(lower is better)* for different models and training losses for 1, 5 and 10 step prediction for the FixedUnobserved Physics environment setting with 3 objects.

		1 Step			5 Steps			10 Steps		
	Model	H@1	MRR	Rec.	H@1	MRR	Rec.	H@1	MRR	Rec.
NLL	AE	98.51 $\pm$0.14	98.85 $\pm$0.09	0.07 $\pm$0.0	84.3 $\pm$1.58	87.34 $\pm$1.18	0.16 $\pm$0.0	58.14 $\pm$2.86	64.57 $\pm$2.51	0.26 $\pm$0.01
	GNN	95.61 $\pm$2.72	97.13 $\pm$2.01	0.13 $\pm$0.02	53.89 $\pm$11.58	62.05 $\pm$10.9	0.32 $\pm$0.05	28.64 $\pm$11.56	36.96 $\pm$12.45	0.44 $\pm$0.07
	Modular	99.48 $\pm$0.25	99.63 $\pm$0.21	0.07 $\pm$0.0	94.13 $\pm$1.31	95.54 $\pm$1.24	0.15 $\pm$0.01	78.18 $\pm$3.17	82.76 $\pm$2.95	0.23 $\pm$0.02
	VAE	74.2 $\pm$0.99	78.72 $\pm$0.98	0.23 $\pm$0.01	22.18 $\pm$0.93	27.69 $\pm$0.91	0.63 $\pm$0.04	5.68 $\pm$0.88	9.0 $\pm$1.0	0.83 $\pm$0.06
NLL Finetuned	AE	99.18 $\pm$0.16	99.37 $\pm$0.11	0.07 $\pm$0.0	91.48 $\pm$1.86	93.19 $\pm$1.41	0.12 $\pm$0.01	73.28 $\pm$3.92	77.99 $\pm$3.36	0.2 $\pm$0.02
	GNN	95.86 $\pm$3.39	97.36 $\pm$2.31	0.06 $\pm$0.0	65.56 $\pm$16.0	71.71 $\pm$14.42	0.13 $\pm$0.03	35.26 $\pm$20.76	41.63 $\pm$20.99	0.26 $\pm$0.07
	Modular	99.85 $\pm$0.12	99.91 $\pm$0.08	0.06 $\pm$0.0	95.04 $\pm$1.85	96.59 $\pm$1.39	0.11 $\pm$0.01	61.72 $\pm$9.28	68.6 $\pm$8.84	0.23 $\pm$0.02
	VAE	54.28 $\pm$4.29	62.55 $\pm$3.59	0.07 $\pm$0.0	10.07 $\pm$3.42	14.12 $\pm$4.4	0.28 $\pm$0.01	1.91 $\pm$1.13	3.3 $\pm$1.72	0.4 $\pm$0.02
Contrastive	AE	92.83 $\pm$12.62	94.9 $\pm$10.1	-	79.39 $\pm$25.3	85.46 $\pm$21.36	-	72.04 $\pm$26.37	80.28 $\pm$22.74	-
	GNN	99.93 $\pm$0.11	99.97 $\pm$0.06	-	96.21 $\pm$6.69	97.41 $\pm$4.64	-	88.83 $\pm$18.69	91.34 $\pm$14.86	-
	Modular	99.86 $\pm$0.07	99.93 $\pm$0.04	-	98.36 $\pm$0.49	98.94 $\pm$0.42	-	93.44 $\pm$2.91	95.63 $\pm$1.94	-

Table 8: Hits at Rank 1 (H@1), Mean Reciprocal Rank (MRR) *(higher is better)* and Reconstruction Error *(lower is better)* for different models and training losses for 1, 5 and 10 step prediction for the FixedUnobserved Physics environment setting with 5 objects.

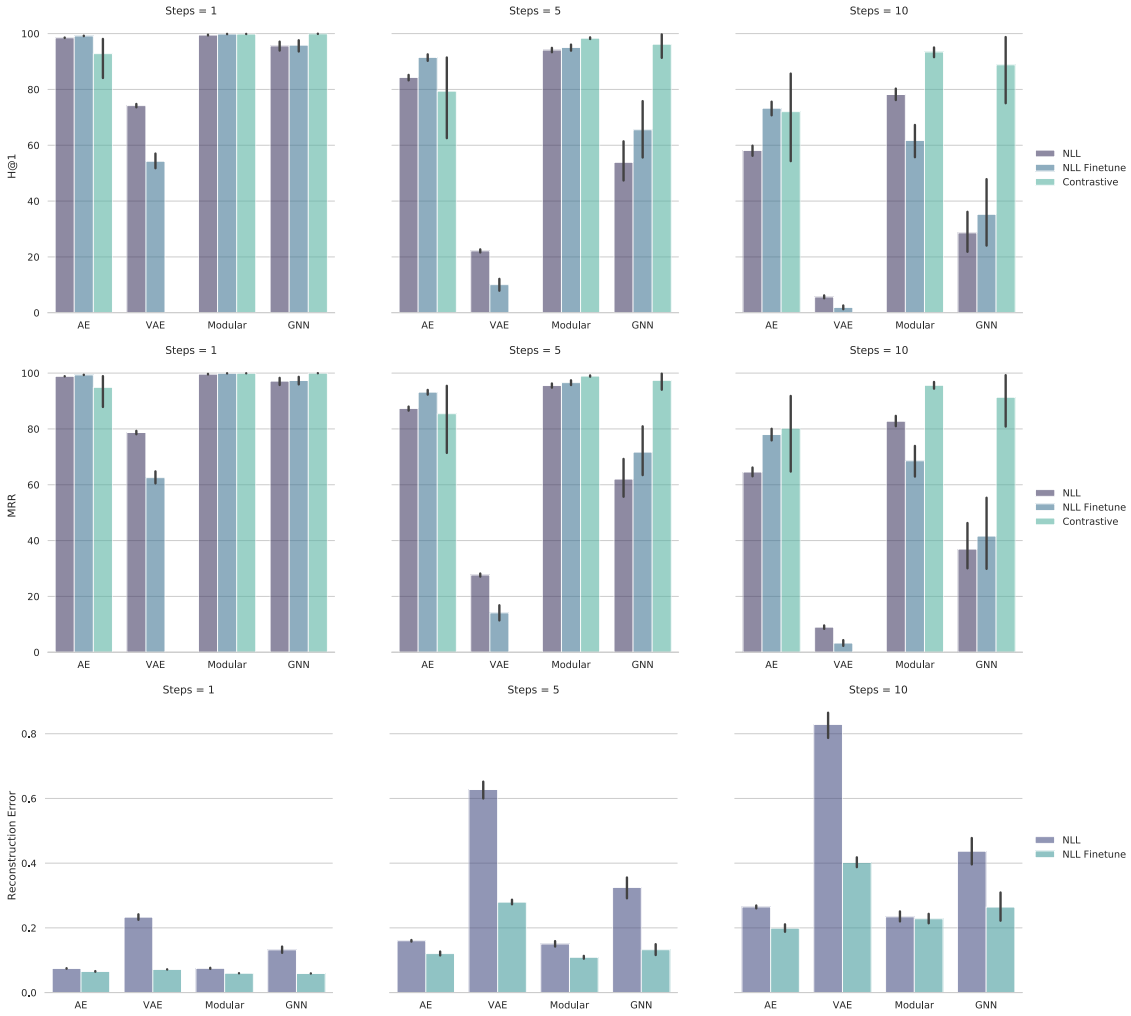

Figure 13: Hits at Rank 1 (H@1), Mean Reciprocal Rank (MRR) *(higher is better)* and Reconstruction Error *(lower is better)* for different models and training losses for 1, 5 and 10 step prediction for the FixedUnobserved Physics environment setting with 5 objects.

	Model	1 Step		5 Steps		10 Steps	
		Reward	Success	Reward	Success	Reward	Success
Baselines	Random Baseline	-0.37	0.22	-1.26	0.01	-1.78	0.00
	Greedy Baseline	0.00	1.00	-0.00	0.99	-0.01	0.98
NLL	AE	$-0.26_{\pm 0.01}$	$0.44_{\pm 0.03}$	$-0.73_{\pm 0.04}$	$0.12_{\pm 0.02}$	$-1.02_{\pm 0.06}$	$0.08_{\pm 0.02}$
	GNN	$-0.34_{\pm 0.02}$	$0.29_{\pm 0.03}$	$-1.04_{\pm 0.07}$	$0.04_{\pm 0.01}$	$-1.49_{\pm 0.1}$	$0.02_{\pm 0.01}$
	Modular	$-0.25_{\pm 0.02}$	$0.46_{\pm 0.04}$	$-0.67_{\pm 0.06}$	$0.15_{\pm 0.02}$	$-0.97_{\pm 0.09}$	$0.08_{\pm 0.02}$
	VAE	$-0.33_{\pm 0.02}$	$0.32_{\pm 0.03}$	$-1.0_{\pm 0.03}$	$0.04_{\pm 0.01}$	$-1.37_{\pm 0.04}$	$0.01_{\pm 0.0}$
NLL Finetuned	AE	$-0.22_{\pm 0.02}$	$0.52_{\pm 0.03}$	$-0.62_{\pm 0.04}$	$0.17_{\pm 0.02}$	$-0.9_{\pm 0.05}$	$0.11_{\pm 0.02}$
	GNN	$-0.36_{\pm 0.06}$	$0.3_{\pm 0.11}$	$-1.06_{\pm 0.24}$	$0.06_{\pm 0.04}$	$-1.57_{\pm 0.33}$	$0.03_{\pm 0.03}$
	Modular	$-0.16_{\pm 0.04}$	$0.64_{\pm 0.08}$	$-0.48_{\pm 0.11}$	$0.27_{\pm 0.09}$	$-0.79_{\pm 0.17}$	$0.15_{\pm 0.06}$
	VAE	$-0.26_{\pm 0.07}$	$0.43_{\pm 0.13}$	$-0.85_{\pm 0.2}$	$0.08_{\pm 0.05}$	$-1.28_{\pm 0.21}$	$0.03_{\pm 0.02}$
Contrastive	AE	$-0.27_{\pm 0.02}$	$0.42_{\pm 0.03}$	$-0.97_{\pm 0.04}$	$0.05_{\pm 0.01}$	$-1.44_{\pm 0.05}$	$0.02_{\pm 0.0}$
	GNN	$-0.11_{\pm 0.17}$	$0.77_{\pm 0.36}$	$-0.36_{\pm 0.52}$	$0.68_{\pm 0.43}$	$-0.5_{\pm 0.72}$	$0.66_{\pm 0.43}$
	Modular	$-0.2_{\pm 0.07}$	$0.54_{\pm 0.13}$	$-0.76_{\pm 0.23}$	$0.13_{\pm 0.08}$	$-1.06_{\pm 0.28}$	$0.07_{\pm 0.05}$

Table 9: Negative Return *(lower is better)* and Success Rate *(higher is better)* for different models and training losses for 1, 5 and 10 step prediction for the Observed Physics environment setting with 3 objects.

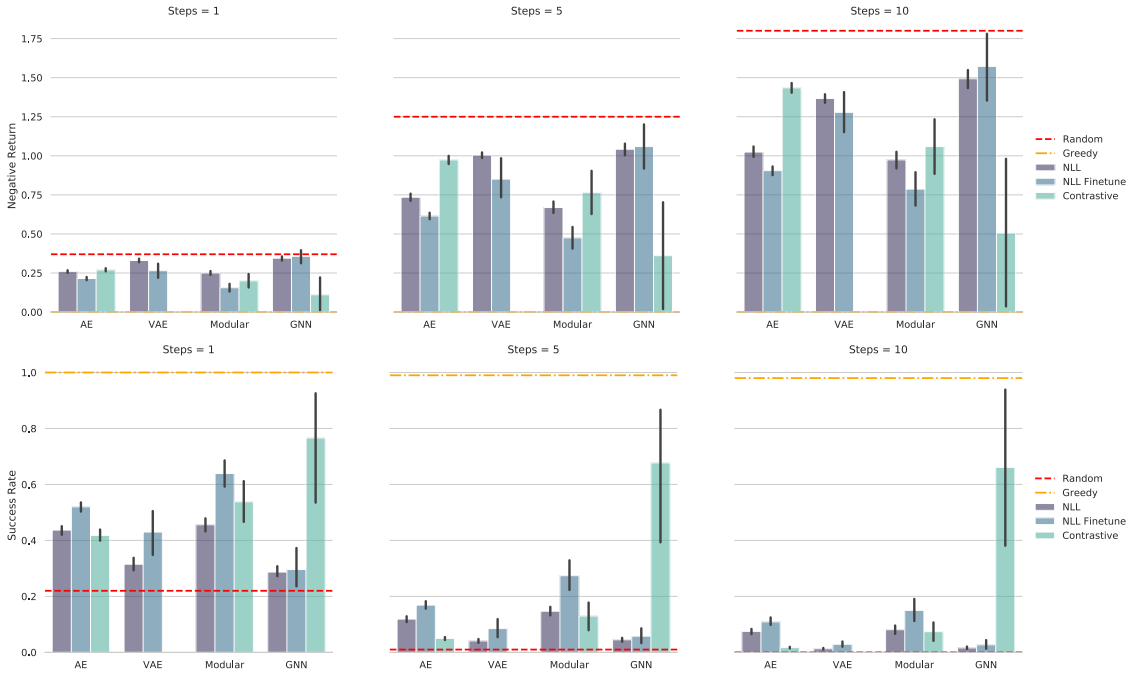

Figure 14: Negative Return *(lower is better)* and Success Rate *(higher is better)* for different models and training losses for 1, 5 and 10 step prediction for the Observed Physics environment setting with 3 objects.

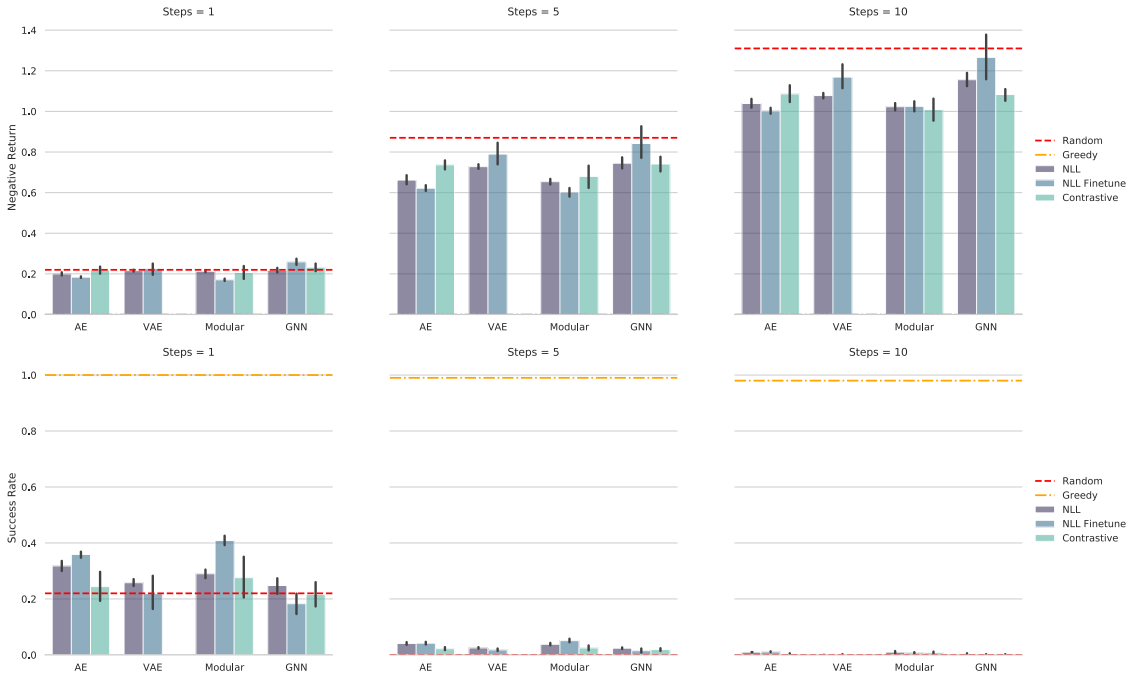

Figure 15: Negative Return *(lower is better)* and Success Rate *(higher is better)* for different models and training losses for 1, 5 and 10 step prediction for the Observed Physics environment setting with 5 objects.

Figure 16: Negative Return *(lower is better)* and Success Rate *(higher is better)* for different models and training losses for 1, 5 and 10 step prediction for the Unobserved Physics environment setting with 3 objects.

Figure 17: Negative Return *(lower is better)* and Success Rate *(higher is better)* for different models and training losses for 1, 5 and 10 step prediction for the Unobserved Physics environment setting with 5 objects.

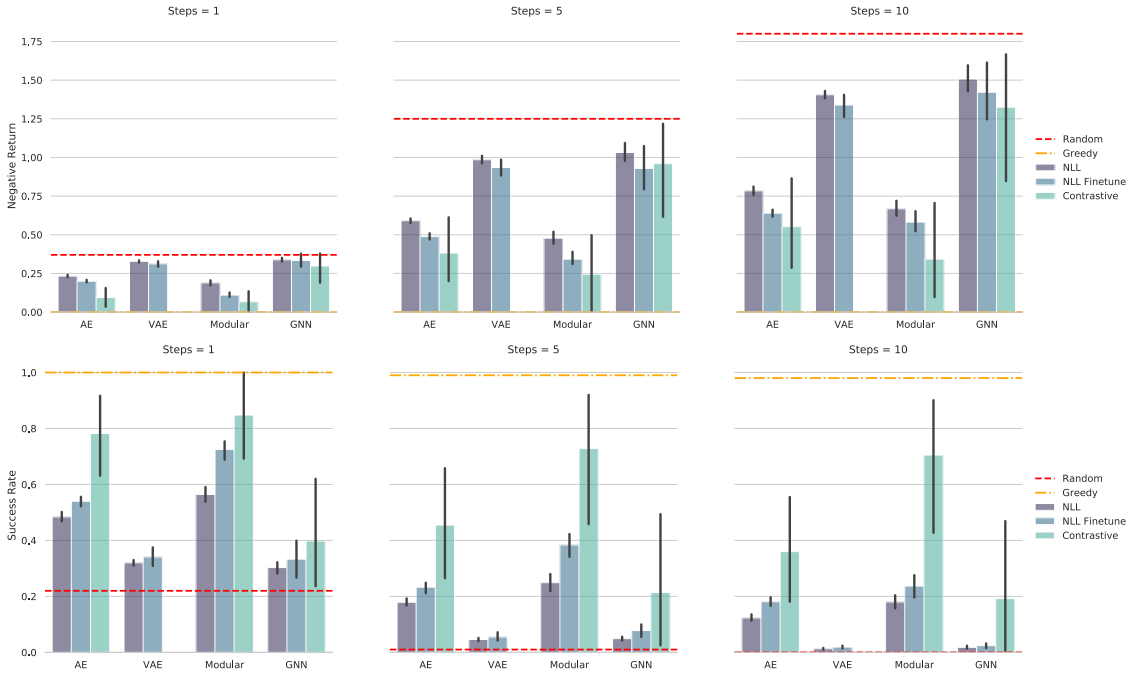

Figure 18: Negative Return *(lower is better)* and Success Rate *(higher is better)* for different models and training losses for 1, 5 and 10 step prediction for the FixedUnobserved Physics environment setting with 3 objects.

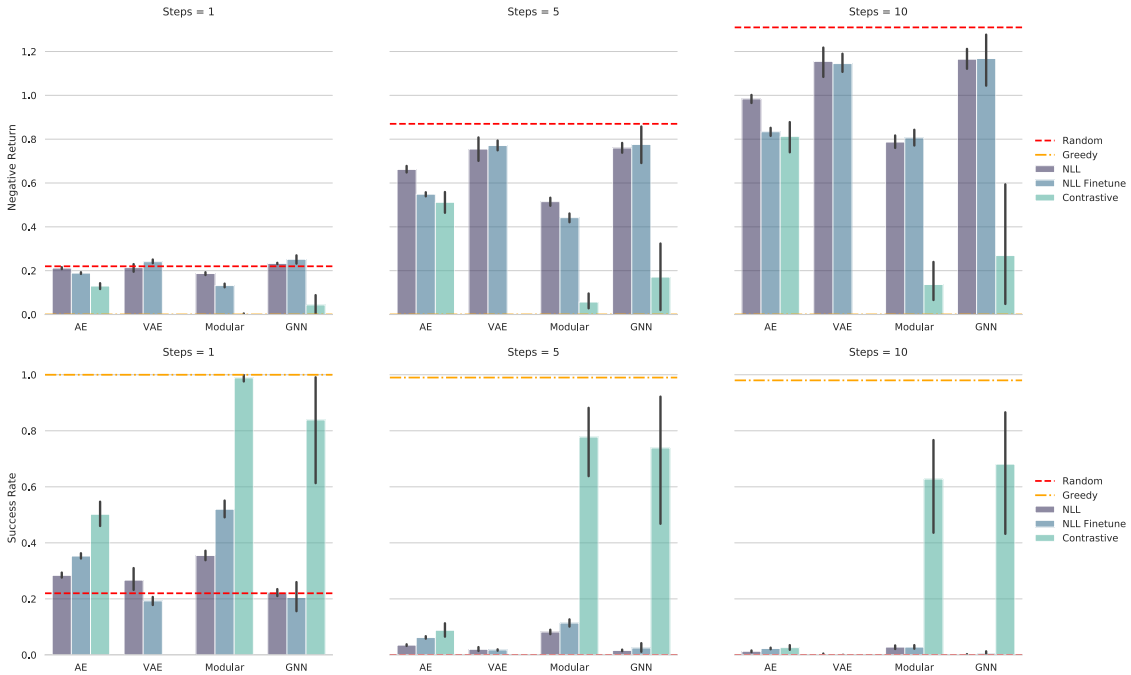

Figure 19: Negative Return *(lower is better)* and Success Rate *(higher is better)* for different models and training losses for 1, 5 and 10 step prediction for the FixedUnobserved Physics environment setting with 5 objects.

Figure 20: Hits at Rank 1 (H@1), Mean Reciprocal Rank (MRR) *(higher is better)* and Reconstruction Error *(lower is better)* for different models and training losses for 1, 5 and 10 step prediction for the Observed Physics environment Zero Shot setting with 3 objects.

		1 Step		5 Steps		10 Steps	
	Model	Reward	Success	Reward	Success	Reward	Success
Baselines	Random Baseline	-0.22	0.22	-0.87	0.00	-1.31	0.00
	Greedy Baseline	0.00	1.00	-0.00	0.99	-0.00	0.98
NLL	AE	-0.2$_{\pm 0.01}$	0.32$_{\pm 0.03}$	-0.66$_{\pm 0.04}$	0.04$_{\pm 0.01}$	-1.04$_{\pm 0.04}$	0.01$_{\pm 0.0}$
	GNN	-0.22$_{\pm 0.02}$	0.25$_{\pm 0.04}$	-0.74$_{\pm 0.04}$	0.02$_{\pm 0.0}$	-1.16$_{\pm 0.06}$	0.0$_{\pm 0.0}$
	Modular	-0.21$_{\pm 0.01}$	0.29$_{\pm 0.03}$	-0.65$_{\pm 0.02}$	0.04$_{\pm 0.01}$	-1.02$_{\pm 0.03}$	0.01$_{\pm 0.01}$
	VAE	-0.22$_{\pm 0.01}$	0.26$_{\pm 0.02}$	-0.73$_{\pm 0.02}$	0.02$_{\pm 0.0}$	-1.08$_{\pm 0.02}$	0.0$_{\pm 0.0}$
NLL Finetuned	AE	-0.18$_{\pm 0.01}$	0.36$_{\pm 0.02}$	-0.62$_{\pm 0.02}$	0.04$_{\pm 0.01}$	-1.0$_{\pm 0.02}$	0.01$_{\pm 0.0}$
	GNN	-0.26$_{\pm 0.03}$	0.18$_{\pm 0.06}$	-0.84$_{\pm 0.12}$	0.02$_{\pm 0.01}$	-1.27$_{\pm 0.18}$	0.0$_{\pm 0.0}$
	Modular	-0.17$_{\pm 0.01}$	0.41$_{\pm 0.03}$	-0.6$_{\pm 0.03}$	0.05$_{\pm 0.01}$	-1.02$_{\pm 0.04}$	0.01$_{\pm 0.0}$
	VAE	-0.23$_{\pm 0.04}$	0.22$_{\pm 0.1}$	-0.79$_{\pm 0.08}$	0.02$_{\pm 0.01}$	-1.17$_{\pm 0.1}$	0.0$_{\pm 0.0}$
Contrastive	AE	-0.22$_{\pm 0.03}$	0.24$_{\pm 0.08}$	-0.74$_{\pm 0.04}$	0.02$_{\pm 0.01}$	-1.09$_{\pm 0.07}$	0.0$_{\pm 0.0}$
	GNN	-0.23$_{\pm 0.03}$	0.22$_{\pm 0.07}$	-0.74$_{\pm 0.06}$	0.02$_{\pm 0.01}$	-1.08$_{\pm 0.05}$	0.0$_{\pm 0.0}$
	Modular	-0.21$_{\pm 0.05}$	0.28$_{\pm 0.12}$	-0.68$_{\pm 0.09}$	0.02$_{\pm 0.02}$	-1.01$_{\pm 0.09}$	0.01$_{\pm 0.01}$

Table 10: Negative Return *(lower is better)* and Success Rate *(higher is better)* for different models and training losses for 1, 5 and 10 step prediction for the Observed Physics environment setting with 5 objects.

Figure 21: Hits at Rank 1 (H@1), Mean Reciprocal Rank (MRR) *(higher is better)* and Reconstruction Error *(lower is better)* for different models and training losses for 1, 5 and 10 step prediction for the Observed Physics environment Zero Shot setting with 5 objects.

		1 Step		5 Steps		10 Steps	
	Model	Reward	Success	Reward	Success	Reward	Success
Baselines	Random Baseline	-0.37	0.22	-1.26	0.01	-1.78	0.00
	Greedy Baseline	0.00	1.00	-0.00	0.99	-0.01	0.98
NLL	AE	-0.31 $\pm$0.01	0.35 $\pm$0.02	-0.95 $\pm$0.02	0.06 $\pm$0.01	-1.39 $\pm$0.04	0.02 $\pm$0.01
	GNN	-0.36 $\pm$0.01	0.27 $\pm$0.01	-1.13 $\pm$0.02	0.03 $\pm$0.0	-1.64 $\pm$0.03	0.01 $\pm$0.0
	Modular	-0.32 $\pm$0.01	0.34 $\pm$0.01	-0.94 $\pm$0.02	0.06 $\pm$0.01	-1.36 $\pm$0.04	0.02 $\pm$0.01
	VAE	-0.37 $\pm$0.01	0.26 $\pm$0.03	-1.06 $\pm$0.06	0.04 $\pm$0.01	-1.48 $\pm$0.05	0.01 $\pm$0.0
NLL Finetuned	AE	-0.26 $\pm$0.02	0.44 $\pm$0.03	-0.83 $\pm$0.06	0.08 $\pm$0.02	-1.23 $\pm$0.07	0.03 $\pm$0.01
	GNN	-0.37 $\pm$0.02	0.26 $\pm$0.03	-1.13 $\pm$0.05	0.03 $\pm$0.01	-1.71 $\pm$0.1	0.01 $\pm$0.01
	Modular	-0.27 $\pm$0.03	0.43 $\pm$0.05	-0.89 $\pm$0.07	0.07 $\pm$0.02	-1.32 $\pm$0.09	0.02 $\pm$0.01
	VAE	-0.39 $\pm$0.03	0.22 $\pm$0.03	-1.18 $\pm$0.07	0.02 $\pm$0.01	-1.61 $\pm$0.09	0.0 $\pm$0.0
Contrastive	AE	-0.31 $\pm$0.02	0.36 $\pm$0.04	-0.96 $\pm$0.04	0.05 $\pm$0.01	-1.36 $\pm$0.05	0.01 $\pm$0.01
	GNN	-0.39 $\pm$0.02	0.2 $\pm$0.04	-1.22 $\pm$0.06	0.02 $\pm$0.01	-1.64 $\pm$0.04	0.0 $\pm$0.0
	Modular	-0.31 $\pm$0.03	0.37 $\pm$0.06	-1.07 $\pm$0.07	0.04 $\pm$0.01	-1.54 $\pm$0.09	0.01 $\pm$0.0

Table 11: Negative Return *(lower is better)* and Success Rate *(higher is better)* for different models and training losses for 1, 5 and 10 step prediction for the Unobserved Physics environment setting with 3 objects.

Figure 22: Plots for Observed Physics Environment with 3 objects. Note that (a) the ranking metric (H@1) does not always correspond to good RL performance. In particular, the ranking metric is good across multiple steps but RL performance generally degrades. (b) and (c) Ranking metric and success rate seem to be a bit negatively correlated with test loss.

H Chemistry Environment

H.1 Detailed Setup

The chemistry environment consists of objects of different shapes and colors. Each object forms a node of a directed acyclic graph. The shapes and positions of the objects are fixed across episodes while the color of each object is sampled from a conditional probability table and depends on the colors of its ancestors.

Considering a set of M objects: $(X_i = \{s_i, c_i, p_i\} \quad \forall i \in \{1, \ldots, M\})$. Here, s_i, c_i and p_i denote the shape, color and position of the object respectively. As mentioned previously, the shapes and the positions are fixed across episodes but different for each object. The color of an object is a categorical variable that can take one of the K possible values. To model the CPT we use an MLP for each object, the input to an object's MLP is the current state of each of its parent nodes and the outputs is a probability distribution over k colors out of which one color is sampled for that object. We can control the skewness of the distribution of each object by controlling the initialization of the

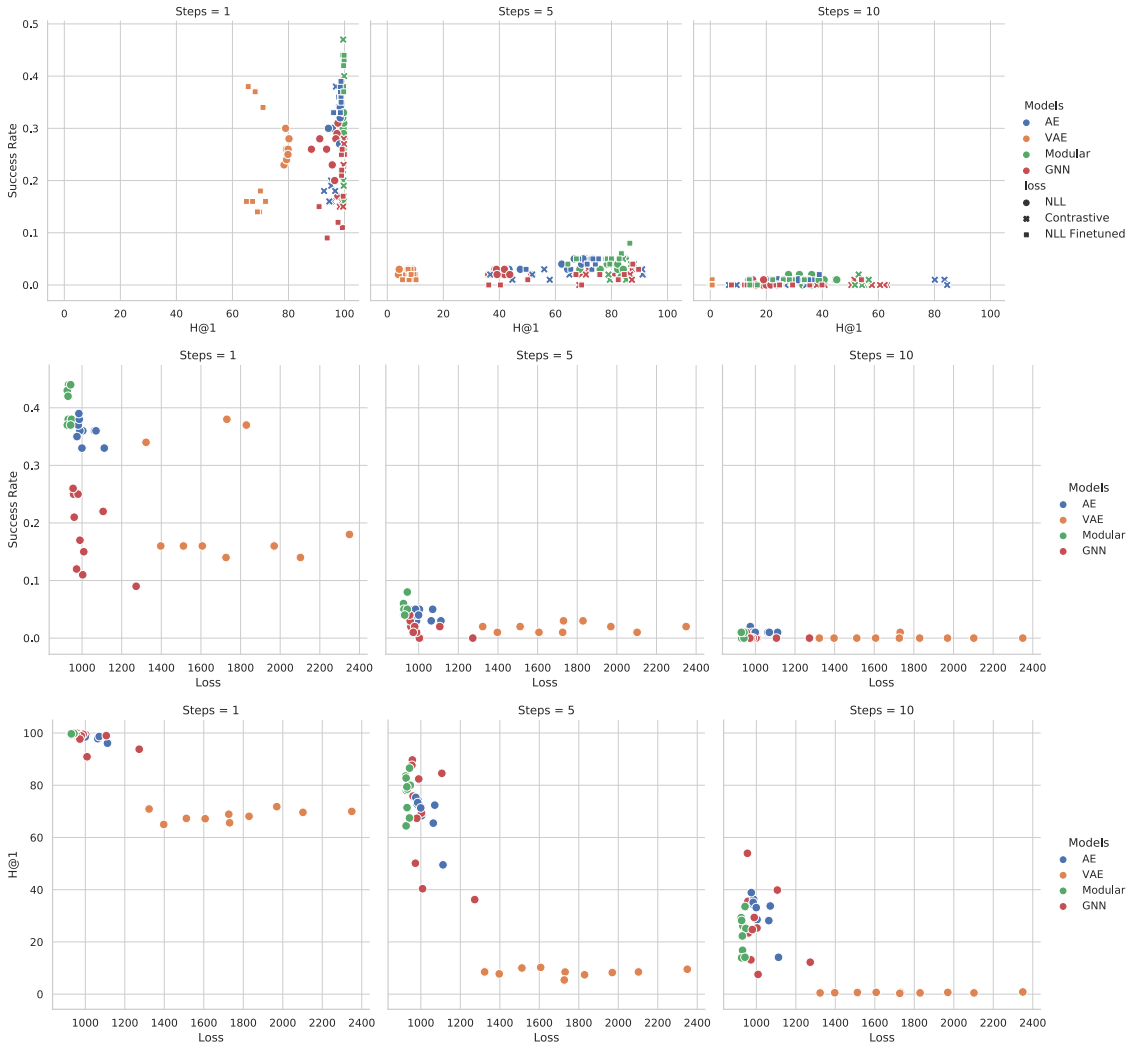

Figure 23: Plots for Observed Physics Environment with 5 objects. Note that (a) the ranking metric (H@1) does not always correspond to good RL performance. In particular, the ranking metric is good across multiple steps but RL performance generally degrades. (b) and (c) Ranking metric and success rate seem to be a bit negatively correlated with test loss.

MLP parameters. It is more hard for a model to learn the correct probability distribution when the distribution is less skewed.

In the chemistry environment, an intervention corresponds to changing the color of an object to a particular color from fixed set of K colors. When an intervention is performed on an object, a new color is sampled for each of its descendants using their respective MLPs as mentioned above. Each object changes its color to the newly sampled color at the same instant.

Note that all our experiments for this environment were run for a setting of 5 objects and 5 colors unless specified otherwise.

H.2 Ranking Loss and Causal Structure

Initially, our vanilla chemistry environment had objects being initialized at random positions per episode while maintaining a fixed causal graph underneath. We call this setting the **dynamic** setting. We noticed that in this case, the ranking metrics were very good but performance on downstream RL task as well as qualitative reconstruction was very poor. On further investigation, we reached

	Model	1 Step		5 Steps		10 Steps	
		Reward	Success	Reward	Success	Reward	Success
Baselines	Random Baseline	-0.22	0.22	-0.87	0.00	-1.31	0.00
	Greedy Baseline	0.00	1.00	-0.00	0.99	-0.00	0.98
NLL	AE	-0.22$_{\pm 0.01}$	0.26$_{\pm 0.03}$	-0.74$_{\pm 0.03}$	0.02$_{\pm 0.01}$	-1.14$_{\pm 0.03}$	0.0$_{\pm 0.0}$
	GNN	-0.22$_{\pm 0.01}$	0.25$_{\pm 0.02}$	-0.76$_{\pm 0.02}$	0.02$_{\pm 0.01}$	-1.19$_{\pm 0.03}$	0.0$_{\pm 0.0}$
	Modular	-0.21$_{\pm 0.01}$	0.28$_{\pm 0.03}$	-0.7$_{\pm 0.03}$	0.02$_{\pm 0.0}$	-1.08$_{\pm 0.04}$	0.01$_{\pm 0.0}$
	VAE	-0.23$_{\pm 0.01}$	0.22$_{\pm 0.02}$	-0.78$_{\pm 0.03}$	0.02$_{\pm 0.01}$	-1.18$_{\pm 0.06}$	0.0$_{\pm 0.0}$
NLL Finetuned	AE	-0.18$_{\pm 0.02}$	0.35$_{\pm 0.04}$	-0.59$_{\pm 0.05}$	0.04$_{\pm 0.01}$	-0.96$_{\pm 0.07}$	0.01$_{\pm 0.0}$
	GNN	-0.23$_{\pm 0.0}$	0.22$_{\pm 0.02}$	-0.8$_{\pm 0.04}$	0.02$_{\pm 0.01}$	-1.28$_{\pm 0.07}$	0.0$_{\pm 0.0}$
	Modular	-0.21$_{\pm 0.01}$	0.28$_{\pm 0.03}$	-0.69$_{\pm 0.04}$	0.03$_{\pm 0.01}$	-1.11$_{\pm 0.07}$	0.01$_{\pm 0.0}$
	VAE	-0.21$_{\pm 0.05}$	0.25$_{\pm 0.11}$	-0.73$_{\pm 0.13}$	0.03$_{\pm 0.01}$	-1.1$_{\pm 0.13}$	0.0$_{\pm 0.0}$
Contrastive	AE	-0.25$_{\pm 0.02}$	0.2$_{\pm 0.06}$	-0.76$_{\pm 0.07}$	0.02$_{\pm 0.01}$	-1.12$_{\pm 0.09}$	0.0$_{\pm 0.0}$
	GNN	-0.21$_{\pm 0.02}$	0.25$_{\pm 0.06}$	-0.71$_{\pm 0.07}$	0.02$_{\pm 0.0}$	-1.08$_{\pm 0.07}$	0.0$_{\pm 0.0}$
	Modular	-0.24$_{\pm 0.02}$	0.2$_{\pm 0.04}$	-0.76$_{\pm 0.05}$	0.02$_{\pm 0.0}$	-1.12$_{\pm 0.06}$	0.0$_{\pm 0.0}$

Table 12: Negative Return *(lower is better)* and Success Rate *(higher is better)* for different models and training losses for 1, 5 and 10 step prediction for the Unobserved Physics environment setting with 5 objects.

	Model	1 Step		5 Steps		10 Steps	
		Reward	Success	Reward	Success	Reward	Success
Baselines	Random Baseline	-0.37	0.22	-1.26	0.01	-1.78	0.00
	Greedy Baseline	0.00	1.00	-0.00	0.99	-0.01	0.98
NLL	AE	-0.23$_{\pm 0.01}$	0.48$_{\pm 0.03}$	-0.59$_{\pm 0.03}$	0.18$_{\pm 0.02}$	-0.78$_{\pm 0.05}$	0.12$_{\pm 0.02}$
	GNN	-0.34$_{\pm 0.02}$	0.3$_{\pm 0.03}$	-1.03$_{\pm 0.09}$	0.05$_{\pm 0.01}$	-1.51$_{\pm 0.14}$	0.02$_{\pm 0.01}$
	Modular	-0.19$_{\pm 0.02}$	0.56$_{\pm 0.04}$	-0.48$_{\pm 0.06}$	0.25$_{\pm 0.05}$	-0.67$_{\pm 0.08}$	0.18$_{\pm 0.04}$
	VAE	-0.33$_{\pm 0.01}$	0.32$_{\pm 0.02}$	-0.98$_{\pm 0.04}$	0.05$_{\pm 0.01}$	-1.4$_{\pm 0.04}$	0.01$_{\pm 0.0}$
NLL Finetuned	AE	-0.2$_{\pm 0.01}$	0.54$_{\pm 0.03}$	-0.49$_{\pm 0.03}$	0.23$_{\pm 0.03}$	-0.64$_{\pm 0.04}$	0.18$_{\pm 0.02}$
	GNN	-0.33$_{\pm 0.07}$	0.33$_{\pm 0.11}$	-0.93$_{\pm 0.22}$	0.08$_{\pm 0.04}$	-1.42$_{\pm 0.3}$	0.02$_{\pm 0.01}$
	Modular	-0.11$_{\pm 0.02}$	0.73$_{\pm 0.05}$	-0.34$_{\pm 0.06}$	0.38$_{\pm 0.07}$	-0.58$_{\pm 0.1}$	0.24$_{\pm 0.06}$
	VAE	-0.31$_{\pm 0.03}$	0.34$_{\pm 0.05}$	-0.94$_{\pm 0.09}$	0.06$_{\pm 0.02}$	-1.34$_{\pm 0.11}$	0.02$_{\pm 0.01}$
Contrastive	AE	-0.09$_{\pm 0.1}$	0.78$_{\pm 0.23}$	-0.38$_{\pm 0.33}$	0.45$_{\pm 0.31}$	-0.55$_{\pm 0.46}$	0.36$_{\pm 0.31}$
	GNN	-0.3$_{\pm 0.15}$	0.4$_{\pm 0.3}$	-0.96$_{\pm 0.48}$	0.21$_{\pm 0.38}$	-1.32$_{\pm 0.65}$	0.19$_{\pm 0.37}$
	Modular	-0.07$_{\pm 0.11}$	0.85$_{\pm 0.24}$	-0.24$_{\pm 0.38}$	0.73$_{\pm 0.41}$	-0.34$_{\pm 0.51}$	0.7$_{\pm 0.43}$

Table 13: Negative Return *(lower is better)* and Success Rate *(higher is better)* for different models and training losses for 1, 5 and 10 step prediction for the FixedUnobserved Physics environment setting with 3 objects.

	Model	1 Step		5 Steps		10 Steps	
		Reward	Success	Reward	Success	Reward	Success
Baselines	Random Baseline	-0.22	0.22	-0.87	0.00	-1.31	0.00
	Greedy Baseline	0.00	1.00	-0.00	0.99	-0.00	0.98
NLL	AE	-0.21$_{\pm 0.01}$	0.28$_{\pm 0.01}$	-0.66$_{\pm 0.02}$	0.04$_{\pm 0.01}$	-0.98$_{\pm 0.03}$	0.01$_{\pm 0.0}$
	GNN	-0.23$_{\pm 0.0}$	0.22$_{\pm 0.02}$	-0.76$_{\pm 0.04}$	0.02$_{\pm 0.0}$	-1.17$_{\pm 0.07}$	0.0$_{\pm 0.0}$
	Modular	-0.19$_{\pm 0.01}$	0.36$_{\pm 0.03}$	-0.51$_{\pm 0.03}$	0.08$_{\pm 0.01}$	-0.79$_{\pm 0.05}$	0.03$_{\pm 0.01}$
	VAE	-0.21$_{\pm 0.03}$	0.27$_{\pm 0.06}$	-0.75$_{\pm 0.09}$	0.02$_{\pm 0.01}$	-1.16$_{\pm 0.1}$	0.0$_{\pm 0.0}$
NLL Finetuned	AE	-0.19$_{\pm 0.01}$	0.35$_{\pm 0.01}$	-0.55$_{\pm 0.02}$	0.06$_{\pm 0.01}$	-0.83$_{\pm 0.03}$	0.02$_{\pm 0.0}$
	GNN	-0.25$_{\pm 0.03}$	0.2$_{\pm 0.08}$	-0.78$_{\pm 0.14}$	0.02$_{\pm 0.03}$	-1.17$_{\pm 0.19}$	0.01$_{\pm 0.01}$
	Modular	-0.13$_{\pm 0.01}$	0.52$_{\pm 0.05}$	-0.44$_{\pm 0.03}$	0.11$_{\pm 0.02}$	-0.81$_{\pm 0.06}$	0.03$_{\pm 0.01}$
	VAE	-0.24$_{\pm 0.01}$	0.19$_{\pm 0.02}$	-0.77$_{\pm 0.04}$	0.02$_{\pm 0.0}$	-1.14$_{\pm 0.07}$	0.0$_{\pm 0.0}$
Contrastive	AE	-0.13$_{\pm 0.02}$	0.5$_{\pm 0.07}$	-0.51$_{\pm 0.08}$	0.09$_{\pm 0.04}$	-0.81$_{\pm 0.11}$	0.03$_{\pm 0.01}$
	GNN	-0.04$_{\pm 0.09}$	0.84$_{\pm 0.3}$	-0.17$_{\pm 0.3}$	0.74$_{\pm 0.36}$	-0.27$_{\pm 0.44}$	0.68$_{\pm 0.34}$
	Modular	-0.0$_{\pm 0.0}$	0.99$_{\pm 0.02}$	-0.06$_{\pm 0.06}$	0.78$_{\pm 0.2}$	-0.14$_{\pm 0.14}$	0.63$_{\pm 0.27}$

Table 14: Negative Return *(lower is better)* and Success Rate *(higher is better)* for different models and training losses for 1, 5 and 10 step prediction for the FixedUnobserved Physics environment setting with 5 objects.

the conclusion that under this setting, a model could do very well under the ranking metrics without learning the causal structure at all.

	Model	1 Step H@1	1 Step MRR	1 Step Rec.	5 Steps H@1	5 Steps MRR	5 Steps Rec.	10 Steps H@1	10 Steps MRR	10 Steps Rec.
NLL	AE	$73.41_{\pm 0.63}$	$78.83_{\pm 0.54}$	$0.1_{\pm 0.0}$	$26.32_{\pm 1.55}$	$31.97_{\pm 1.53}$	$0.31_{\pm 0.01}$	$10.73_{\pm 1.19}$	$14.47_{\pm 1.32}$	$0.45_{\pm 0.02}$
	GNN	$57.06_{\pm 1.49}$	$65.9_{\pm 1.26}$	$0.15_{\pm 0.01}$	$12.08_{\pm 1.15}$	$18.33_{\pm 1.49}$	$0.4_{\pm 0.04}$	$3.6_{\pm 0.67}$	$6.97_{\pm 1.11}$	$0.5_{\pm 0.05}$
	Modular	$71.67_{\pm 1.47}$	$77.8_{\pm 1.21}$	$0.12_{\pm 0.0}$	$26.7_{\pm 4.2}$	$33.58_{\pm 4.68}$	$0.31_{\pm 0.02}$	$10.84_{\pm 2.89}$	$15.38_{\pm 3.71}$	$0.42_{\pm 0.03}$
	VAE	$43.78_{\pm 1.57}$	$55.48_{\pm 1.93}$	$0.29_{\pm 0.02}$	$1.59_{\pm 0.1}$	$3.18_{\pm 0.1}$	$1.09_{\pm 0.12}$	$0.12_{\pm 0.03}$	$0.46_{\pm 0.04}$	$1.23_{\pm 0.14}$
NLL Finetuned	AE	$73.78_{\pm 1.86}$	$79.35_{\pm 1.73}$	$0.09_{\pm 0.01}$	$28.37_{\pm 1.55}$	$33.98_{\pm 1.65}$	$0.29_{\pm 0.01}$	$12.4_{\pm 0.88}$	$16.09_{\pm 1.04}$	$0.43_{\pm 0.01}$
	GNN	$66.42_{\pm 7.06}$	$72.43_{\pm 6.67}$	$0.1_{\pm 0.01}$	$18.42_{\pm 7.28}$	$24.18_{\pm 8.74}$	$0.24_{\pm 0.02}$	$3.24_{\pm 1.99}$	$5.26_{\pm 2.78}$	$0.36_{\pm 0.03}$
	Modular	$77.33_{\pm 1.83}$	$82.91_{\pm 1.67}$	$0.11_{\pm 0.01}$	$35.97_{\pm 6.75}$	$43.71_{\pm 7.17}$	$0.24_{\pm 0.01}$	$15.73_{\pm 6.19}$	$21.38_{\pm 7.48}$	$0.34_{\pm 0.02}$
	VAE	$62.8_{\pm 14.23}$	$71.95_{\pm 12.27}$	$0.09_{\pm 0.01}$	$9.77_{\pm 6.48}$	$14.44_{\pm 8.41}$	$0.4_{\pm 0.05}$	$0.69_{\pm 0.49}$	$1.63_{\pm 1.02}$	$0.6_{\pm 0.06}$
Contrastive	AE	$72.16_{\pm 1.31}$	$78.78_{\pm 1.07}$	-	$33.23_{\pm 5.11}$	$45.72_{\pm 4.26}$	-	$18.92_{\pm 4.56}$	$31.02_{\pm 5.04}$	-
	GNN	$92.19_{\pm 5.86}$	$94.89_{\pm 4.05}$	-	$61.6_{\pm 19.42}$	$69.77_{\pm 17.93}$	-	$44.51_{\pm 21.94}$	$53.42_{\pm 22.27}$	-
	Modular	$85.03_{\pm 1.73}$	$88.08_{\pm 1.88}$	-	$58.26_{\pm 3.25}$	$65.85_{\pm 3.68}$	-	$45.83_{\pm 3.15}$	$54.69_{\pm 3.21}$	-

Table 15: Hits at Rank 1 (H@1), Mean Reciprocal Rank (MRR) *(higher is better)* and Reconstruction Error *(lower is better)* for different models and training losses for 1, 5 and 10 step prediction for the Observed Physics environment Zero Shot setting with 3 objects.

	Model	1 Step H@1	1 Step MRR	1 Step Rec.	5 Steps H@1	5 Steps MRR	5 Steps Rec.	10 Steps H@1	10 Steps MRR	10 Steps Rec.
NLL	AE	$85.81_{\pm 1.18}$	$89.11_{\pm 1.05}$	$0.15_{\pm 0.0}$	$32.64_{\pm 2.82}$	$39.22_{\pm 2.92}$	$0.41_{\pm 0.01}$	$10.2_{\pm 1.74}$	$14.34_{\pm 2.0}$	$0.58_{\pm 0.02}$
	GNN	$94.67_{\pm 2.05}$	$96.86_{\pm 1.35}$	$0.2_{\pm 0.0}$	$39.49_{\pm 3.03}$	$48.71_{\pm 3.35}$	$0.49_{\pm 0.05}$	$17.39_{\pm 2.85}$	$24.61_{\pm 3.59}$	$0.65_{\pm 0.06}$
	Modular	$95.68_{\pm 1.94}$	$97.14_{\pm 1.5}$	$0.16_{\pm 0.0}$	$51.19_{\pm 6.06}$	$59.25_{\pm 6.13}$	$0.42_{\pm 0.01}$	$18.94_{\pm 4.4}$	$25.58_{\pm 5.39}$	$0.58_{\pm 0.02}$
	VAE	$79.8_{\pm 0.66}$	$85.83_{\pm 0.54}$	$0.35_{\pm 0.01}$	$4.83_{\pm 1.62}$	$8.52_{\pm 2.25}$	$1.68_{\pm 0.1}$	$0.23_{\pm 0.07}$	$0.76_{\pm 0.18}$	$2.26_{\pm 0.15}$
NLL Finetuned	AE	$86.52_{\pm 0.32}$	$89.83_{\pm 0.29}$	$0.15_{\pm 0.0}$	$36.33_{\pm 2.52}$	$43.14_{\pm 2.41}$	$0.39_{\pm 0.01}$	$12.12_{\pm 1.92}$	$16.72_{\pm 2.29}$	$0.56_{\pm 0.02}$
	GNN	$96.29_{\pm 1.99}$	$97.27_{\pm 1.57}$	$0.15_{\pm 0.01}$	$51.4_{\pm 9.48}$	$58.06_{\pm 9.27}$	$0.4_{\pm 0.06}$	$13.22_{\pm 5.04}$	$17.9_{\pm 6.0}$	$0.64_{\pm 0.14}$
	Modular	$96.5_{\pm 1.23}$	$97.55_{\pm 0.94}$	$0.16_{\pm 0.02}$	$49.09_{\pm 6.16}$	$56.4_{\pm 6.05}$	$0.43_{\pm 0.08}$	$10.47_{\pm 2.52}$	$14.57_{\pm 3.2}$	$0.69_{\pm 0.16}$
	VAE	$65.76_{\pm 1.61}$	$72.93_{\pm 1.24}$	$0.12_{\pm 0.0}$	$7.39_{\pm 0.77}$	$11.18_{\pm 0.95}$	$0.77_{\pm 0.03}$	$0.43_{\pm 0.06}$	$1.02_{\pm 0.1}$	$1.11_{\pm 0.06}$
Contrastive	AE	$93.92_{\pm 2.23}$	$95.64_{\pm 2.18}$	-	$58.72_{\pm 13.26}$	$68.87_{\pm 10.01}$	-	$34.58_{\pm 21.13}$	$45.31_{\pm 20.27}$	-
	GNN	$99.63_{\pm 0.37}$	$99.8_{\pm 0.21}$	-	$82.16_{\pm 8.14}$	$87.05_{\pm 6.6}$	-	$55.34_{\pm 12.14}$	$64.19_{\pm 11.66}$	-
	Modular	$99.84_{\pm 0.11}$	$99.91_{\pm 0.06}$	-	$86.88_{\pm 3.19}$	$91.02_{\pm 2.51}$	-	$55.64_{\pm 5.68}$	$65.58_{\pm 5.57}$	-

Table 16: Hits at Rank 1 (H@1), Mean Reciprocal Rank (MRR) *(higher is better)* and Reconstruction Error *(lower is better)* for different models and training losses for 1, 5 and 10 step prediction for the Observed Physics environment Zero Shot setting with 5 objects.

If the encoder learns to encode the positions and shapes of different objects, then it already does a great job at ranking. This is because ranking is done with respect to a large buffer of encoded states and since objects are randomly initialized per episode, there is very little probability that two encoded states share the exact same object shapes and positions. Thus, as long as the encoder and the transition function exploit the fact that two encoded states should be close by *iff* they have the same objects in the same positions, then it would do very well on the ranking metrics. Note that in

	Success Rate		
Model	From Scratch	Freezed	Finetuned
AE	$0.284_{\pm 0.003}$	$0.293_{\pm 0.003}$	$0.283_{\pm 0.01}$
VAE	$0.28_{\pm 0.006}$	$0.281_{\pm 0.003}$	$0.287_{\pm 0.001}$
Modular	$0.284_{\pm 0.013}$	$0.285_{\pm 0.01}$	$0.317_{\pm 0.026}$
GNN	$0.281_{\pm 0.003}$	$0.284_{\pm 0.009}$	$0.292_{\pm 0.004}$

Table 17: In this table we show the performance of PPO on the observed physics environment. The results indicates that using a pretrained encoder (by either freezing or finetuning the encoder parameters) outperforms the model trained from scratch in all cases, with the finetuned modular networks outperforming all other models. This is an indication that the representations learned by such models help to improve downstream RL performance, even for model-free RL algorithms.

the above argument, the model had a way of ranking well without even learning anything about the edges in the graph, i.e. the structure of interactions between the objects.

To alleviate this problem, we decided to keep the positions of the objects fixed across episodes too. We call this setting the **static** setting. This means that models will not be able to perform well on ranking metrics by just encoding the positions or shapes of the objects (since they are now shared across episodes). The only way to do well on ranking metrics then is to learn the underlying causal structure. We immediately saw a plummet in ranking metrics that confirmed our suspicions that the models were not able to learn the underlying causal structure.

For a demonstration of the mentioned problem refer to Figure 27. In the figure we can see that for the dynamic setting, models achieve a much higher score on the ranking metrics (H@1 and MRR) as compared to the static setting while doing much worse on the downstream RL task as compared the static setting. This further reinforces the importance of using downstream RL tasks for evaluation.

This also shows that inferring the causal graph even in the case of small graphs is a complex problem that current models are not able to solve well. We believe that the existence of this suite of environments provides a platform for extensive study of causality in world models.

H.3 Experimental Results

We perform ablation studies on the chemistry environment with varying factors in the underlying causal graph to study how these factors impact learning. We summarize our findings below -

- It is easier for models to learn the right causal structure when the cause-effect chains are short. For eg., all models perform much better (under all metrics) on the *collider* graph where cause-effect length can be at-most one as opposed to chain and full graph where the cause-effect length is longer (refer to Figure 24 and Table 18)
- *Modular Models* generally perform better than *Graph Neural Networks (GNNs)* when trained using NLL loss because the former can encode *higher-order interactions* while the latter only encodes *pairwise interactions* (refer to Figure 24 and Table 18).
- While models trained on the *dynamic* chemistry environment perform very well on ranking metrics, they don't do well on the downstream RL task. This is because these models don't actually learn the right causal structure but only encode the visual aspects of the particular episode such as shapes and positions. To further investigate this, we decided to keep the objects *stationary*. We saw that the ranking metrics immediately suffer by a large margin because the models couldn't cheat by just encoding the visual details and not the causal structure (refer to Appendix H.2 and Figure 27 for details).
- *Increased stochasticity (entropy)* of the conditional probability tables (CPTs) make it harder for the models to learn (refer to Figure 25). In the figure, we can see that almost all models generally perform better on less stochastic (more skewed) data as compared to more stochastic (less skewed) data.
- Modular models outperform all other models on the downstream RL task (refer to Figures 7 and 26 and Table 19) for all settings(i.e different graphs and number of steps) due to their ability to encode *higher-order interaction* which monolithic models like AEs and VAEs cannot do while Graph Neural Networks(GNNs) only en *pairwise interactions*. We also report 2 baselines *random* and *optimal* as described in Appendix E.2

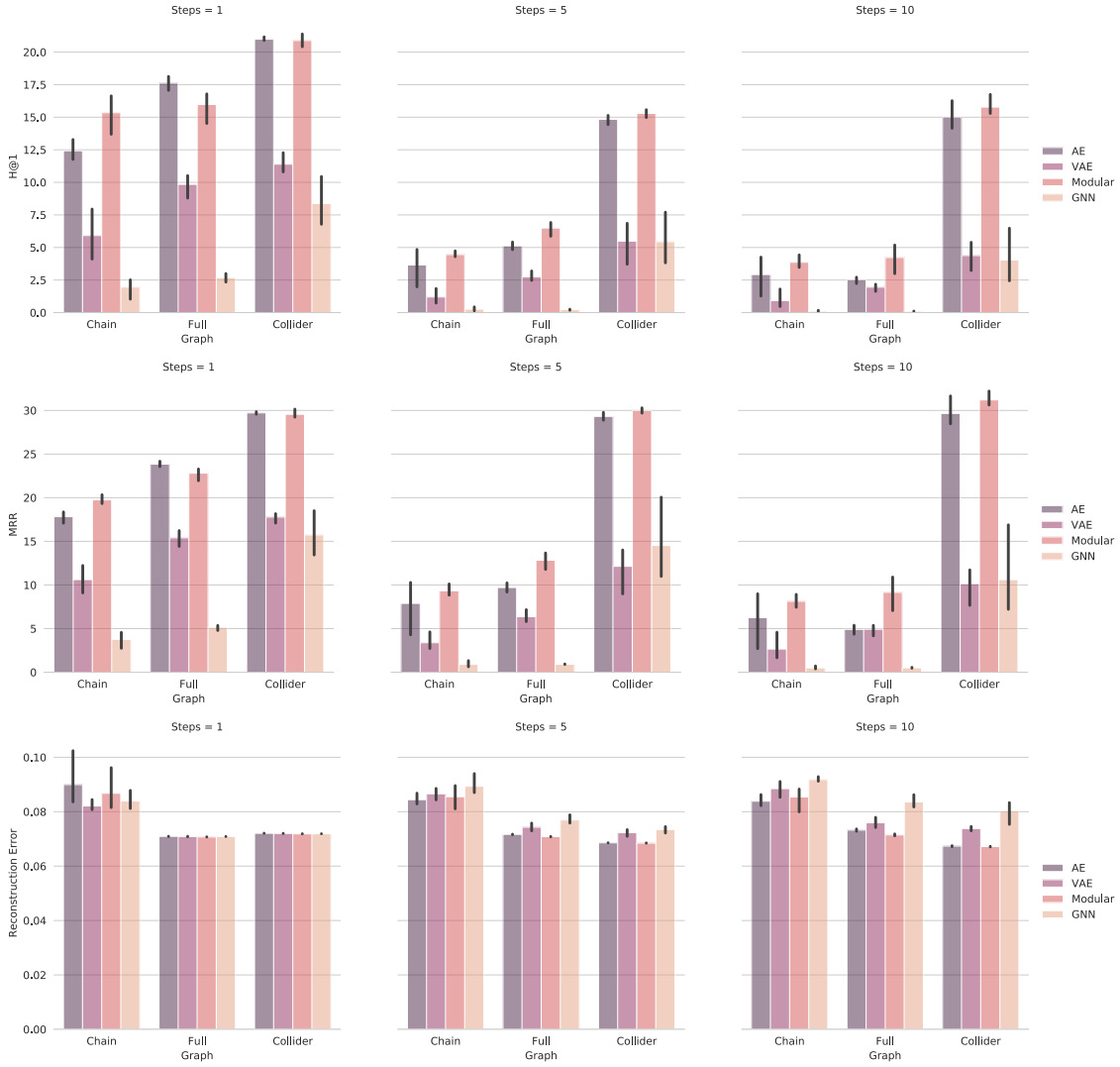

Figure 24: Hits at Rank 1 (H@1), Mean Reciprocal Rank (MRR) *(higher is better)* and Reconstruction Error *(lower is better)* for different models trained using NLL Loss for 1, 5 and 10 step prediction for the vanilla chemistry environment with 5 objects and 5 colors.

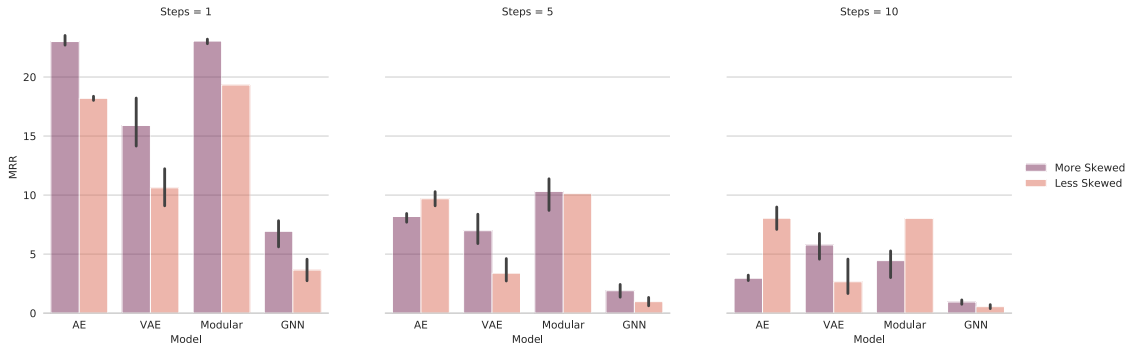

Figure 25: H@1 performance of models for data generated at different levels of skewness(stochasticity) for the chain graph. As we see almost all models perform better on more skewed data as the data uncertainty is less on more skewed data as compared to less skewed data.

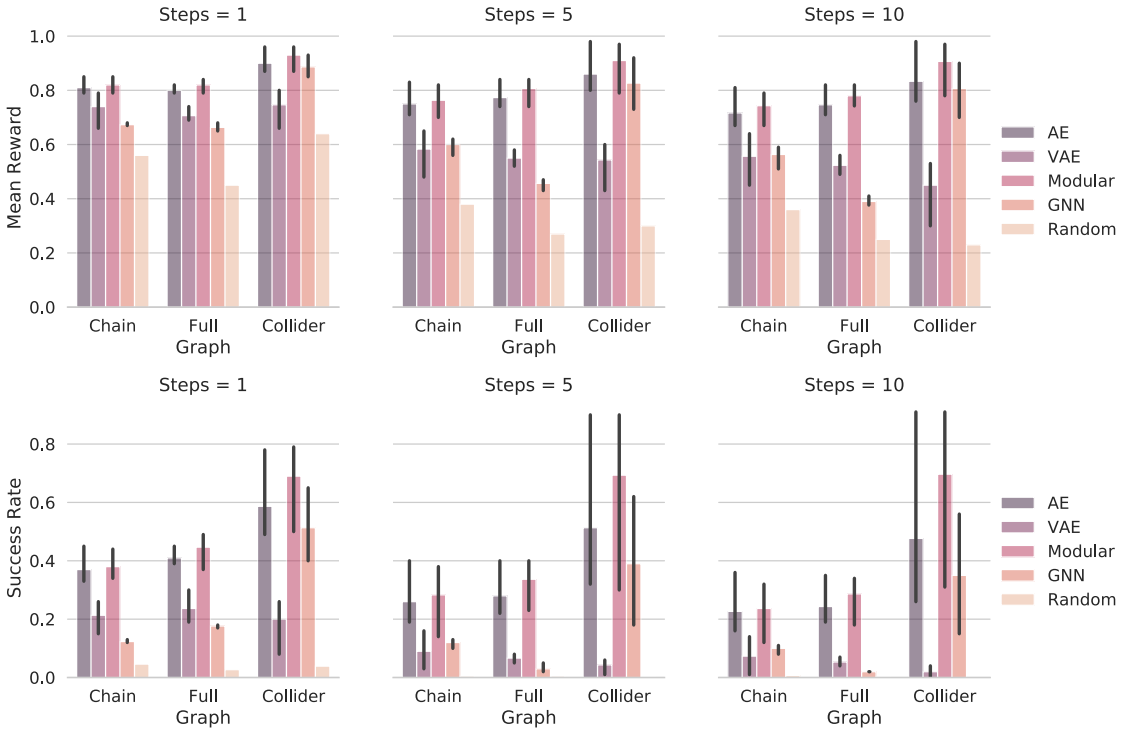

Figure 26: Mean reward and success rate for models trained on the chemistry environment with 5 objects and 5 colors. Modular models outperform all other models in almost all cases which shows that introducing structure in the form of modularity is an important inductive bias for learning causal models.

Graph Type	Model	1 Step H@1	1 Step MRR	1 Step Rec.	5 Steps H@1	5 Steps MRR	5 Steps Rec.	10 Steps H@1	10 Steps MRR	10 Steps Rec.
Chain NLL	AE	16.937$_{\pm 0.386}$	23.007$_{\pm 0.133}$	0.07$_{\pm 0.0}$	4.433$_{\pm 0.023}$	8.187$_{\pm 0.118}$	0.073$_{\pm 0.0}$	1.48$_{\pm 0.04}$	2.957$_{\pm 0.063}$	0.076$_{\pm 0.0}$
	VAE	10.293$_{\pm 2.711}$	15.897$_{\pm 2.927}$	0.071$_{\pm 0.0}$	2.987$_{\pm 0.282}$	6.983$_{\pm 1.079}$	0.075$_{\pm 0.0}$	2.19$_{\pm 0.184}$	5.78$_{\pm 0.821}$	0.076$_{\pm 0.0}$
	Modular	16.863$_{\pm 0.135}$	23.047$_{\pm 0.027}$	0.07$_{\pm 0.0}$	5.317$_{\pm 0.249}$	10.31$_{\pm 1.343}$	0.072$_{\pm 0.0}$	2.04$_{\pm 0.259}$	4.45$_{\pm 1.043}$	0.074$_{\pm 0.0}$
	GNN	3.587$_{\pm 0.412}$	6.93$_{\pm 0.91}$	0.07$_{\pm 0.0}$	0.617$_{\pm 0.05}$	1.9$_{\pm 0.195}$	0.076$_{\pm 0.0}$	0.257$_{\pm 0.002}$	0.947$_{\pm 0.023}$	0.079$_{\pm 0.0}$
Full NLL	AE	17.62$_{\pm 0.192}$	23.85$_{\pm 0.065}$	0.071$_{\pm 0.0}$	5.127$_{\pm 0.058}$	9.707$_{\pm 0.184}$	0.072$_{\pm 0.0}$	2.527$_{\pm 0.045}$	4.913$_{\pm 0.177}$	0.073$_{\pm 0.0}$
	VAE	9.847$_{\pm 0.572}$	15.407$_{\pm 0.559}$	0.071$_{\pm 0.0}$	2.747$_{\pm 0.104}$	6.363$_{\pm 0.342}$	0.074$_{\pm 0.0}$	1.957$_{\pm 0.056}$	4.927$_{\pm 0.289}$	0.076$_{\pm 0.0}$
	Modular	15.977$_{\pm 1.066}$	22.813$_{\pm 0.374}$	0.071$_{\pm 0.0}$	6.493$_{\pm 0.209}$	12.837$_{\pm 0.62}$	0.071$_{\pm 0.0}$	4.233$_{\pm 0.848}$	9.157$_{\pm 2.529}$	0.071$_{\pm 0.0}$
	GNN	2.68$_{\pm 0.073}$	5.15$_{\pm 0.069}$	0.071$_{\pm 0.0}$	0.23$_{\pm 0.001}$	0.913$_{\pm 0.001}$	0.077$_{\pm 0.0}$	0.103$_{\pm 0.001}$	0.503$_{\pm 0.002}$	0.084$_{\pm 0.0}$
Collider NLL	AE	20.993$_{\pm 0.016}$	29.723$_{\pm 0.014}$	0.072$_{\pm 0.0}$	14.84$_{\pm 0.09}$	29.32$_{\pm 0.135}$	0.069$_{\pm 0.0}$	15.01$_{\pm 0.829}$	29.657$_{\pm 2.029}$	0.067$_{\pm 0.0}$
	VAE	9.847$_{\pm 0.572}$	15.407$_{\pm 0.559}$	0.071$_{\pm 0.0}$	2.747$_{\pm 0.104}$	6.363$_{\pm 0.342}$	0.074$_{\pm 0.0}$	1.957$_{\pm 0.056}$	4.927$_{\pm 0.289}$	0.076$_{\pm 0.0}$
	Modular	20.89$_{\pm 0.16}$	29.563$_{\pm 0.173}$	0.072$_{\pm 0.0}$	15.297$_{\pm 0.063}$	29.99$_{\pm 0.062}$	0.068$_{\pm 0.0}$	15.78$_{\pm 0.47}$	31.21$_{\pm 0.515}$	0.067$_{\pm 0.0}$
	GNN	8.377$_{\pm 2.358}$	15.737$_{\pm 4.398}$	0.072$_{\pm 0.0}$	5.443$_{\pm 2.729}$	14.527$_{\pm 15.714}$	0.073$_{\pm 0.0}$	4.04$_{\pm 3.073}$	10.607$_{\pm 20.141}$	0.08$_{\pm 0.0}$

Table 18: Hits at Rank 1 (H@1), Mean Reciprocal Rank (MRR) *(higher is better)* and Reconstruction Error *(lower is better)* for different models trained using **NLL loss** for 1, 5 and 10 step prediction for the vanilla chemistry environment with 5 objects and 5 colors.

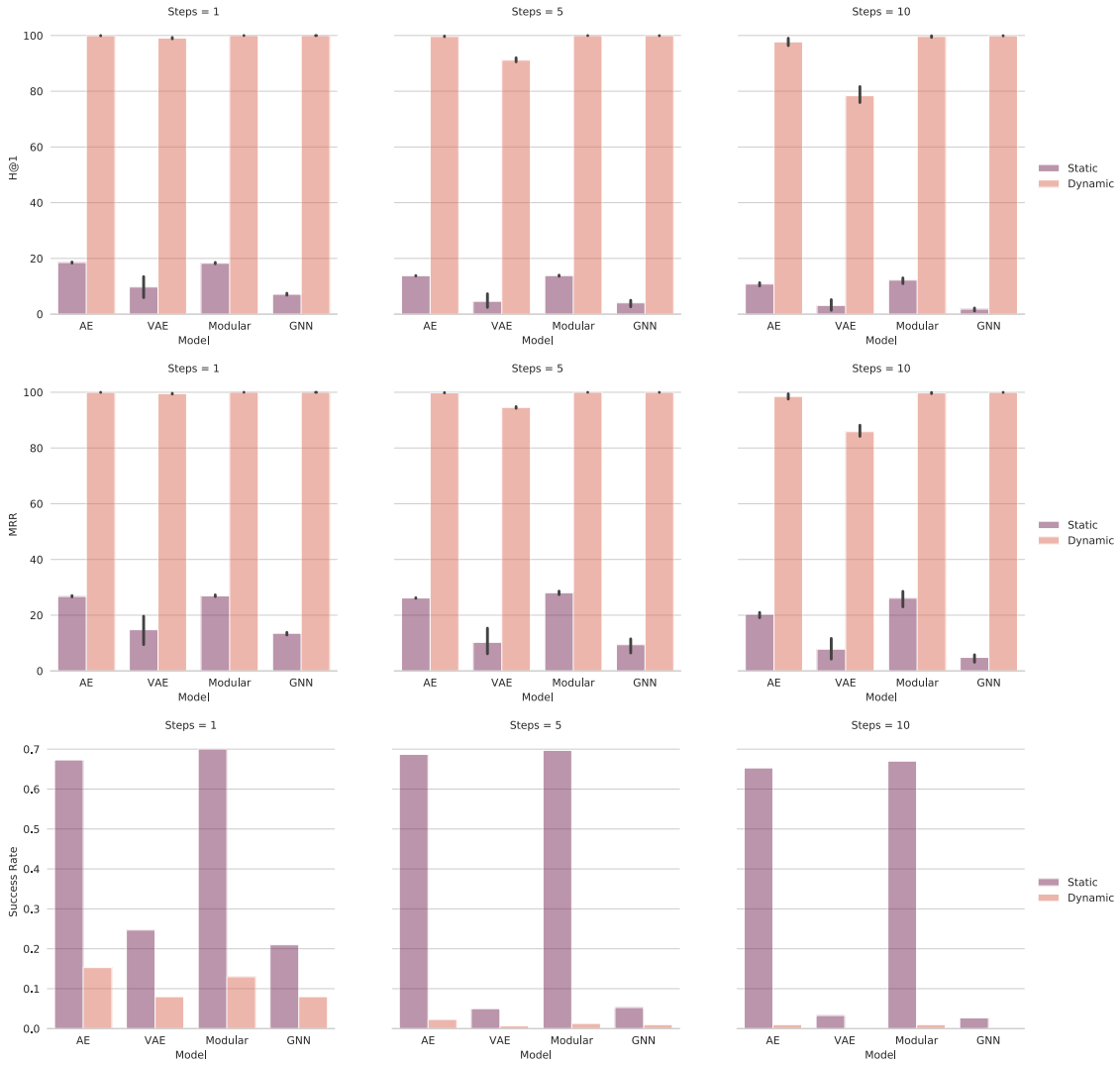

Figure 27: This figure compares the performance of *static* and *dynamic* setting of the chemistry environment. We can see that for the dynamic setting even though the models achieve almost perfect performance on the ranking losses(H@1 and MRR) as compared to the static setting, their performance on the RL task is extremely low as compared to the static setting. This shows that the ranking losses are not an accurate indicator for model performance. For a description of static and dynamic setting see Appendix H.2. These experiments were run for collider graph.

Graph Type	Model	1 Step		5 Steps		10 Steps	
		Mean Reward	Success	Mean Reward	Success	Mean Reward	Success
Chain	Random	0.56	0.046	0.38	0.005	0.36 0.007	
	Optimal	0.86	0.52	0.83	0.39	0.16	0.38
	AE	$0.81_{\pm 0.001}$	$0.37_{\pm 0.003}$	$0.75_{\pm 0.003}$	$0.26_{\pm 0.01}$	$0.717_{\pm 0.004}$	$0.227_{\pm 0.009}$
	VAE	$0.74_{\pm 0.003}$	$0.213_{\pm 0.002}$	$0.583_{\pm 0.005}$	$0.09_{\pm 0.003}$	$0.557_{\pm 0.006}$	$0.073_{\pm 0.003}$
	Modular	$0.82_{\pm 0.001}$	$0.38_{\pm 0.002}$	$0.763_{\pm 0.002}$	$0.283_{\pm 0.011}$	$0.743_{\pm 0.003}$	$0.237_{\pm 0.007}$
	GNN	$0.673_{\pm 0.0}$	$0.123_{\pm 0.0}$	$0.6_{\pm 0.001}$	$0.12_{\pm 0.0}$	$0.563_{\pm 0.001}$	$0.1_{\pm 0.0}$
Full	Random	0.45	0.027	0.27	0.005	0.25	0.004
	Optimal	0.79	0.44	0.737	0.275	0.72	0.24
	AE	$0.8_{\pm 0.0}$	$0.41_{\pm 0.001}$	$0.773_{\pm 0.002}$	$0.28_{\pm 0.007}$	$0.747_{\pm 0.003}$	$0.243_{\pm 0.006}$
	VAE	$0.707_{\pm 0.001}$	$0.237_{\pm 0.002}$	$0.55_{\pm 0.001}$	$0.067_{\pm 0.0}$	$0.523_{\pm 0.001}$	$0.053_{\pm 0.0}$
	Modular	$0.82_{\pm 0.0}$	$0.447_{\pm 0.003}$	$0.807_{\pm 0.002}$	$0.337_{\pm 0.006}$	$0.78_{\pm 0.002}$	$0.287_{\pm 0.006}$
	GNN	$0.663_{\pm 0.0}$	$0.177_{\pm 0.0}$	$0.457_{\pm 0.0}$	$0.03_{\pm 0.0}$	$0.39_{\pm 0.0}$	$0.02_{\pm 0.0}$
Collider	Random	0.45	0.23	0.27	0.005	0.25	0.004
	Optimal	0.95	0.75	0.94	0.733	0.96	0.80
	AE	$0.9_{\pm 0.002}$	$0.587_{\pm 0.019}$	$0.86_{\pm 0.007}$	$0.513_{\pm 0.075}$	$0.833_{\pm 0.011}$	$0.477_{\pm 0.094}$
	VAE	$0.747_{\pm 0.004}$	$0.2_{\pm 0.007}$	$0.543_{\pm 0.006}$	$0.043_{\pm 0.001}$	$0.45_{\pm 0.011}$	$0.02_{\pm 0.0}$
	Modular	$0.93_{\pm 0.002}$	$0.69_{\pm 0.018}$	$0.91_{\pm 0.007}$	$0.693_{\pm 0.077}$	$0.907_{\pm 0.008}$	$0.697_{\pm 0.075}$
	GNN	$0.887_{\pm 0.001}$	$0.513_{\pm 0.011}$	$0.827_{\pm 0.006}$	$0.39_{\pm 0.032}$	$0.807_{\pm 0.007}$	$0.35_{\pm 0.028}$

Table 19: Mean reward and Success rate (*higher is better*) for 1, 5 and 10 step for the vanilla setting of the chemistry environment with 5 objects and 5 colors. This table uses models trained using **NLL loss**.

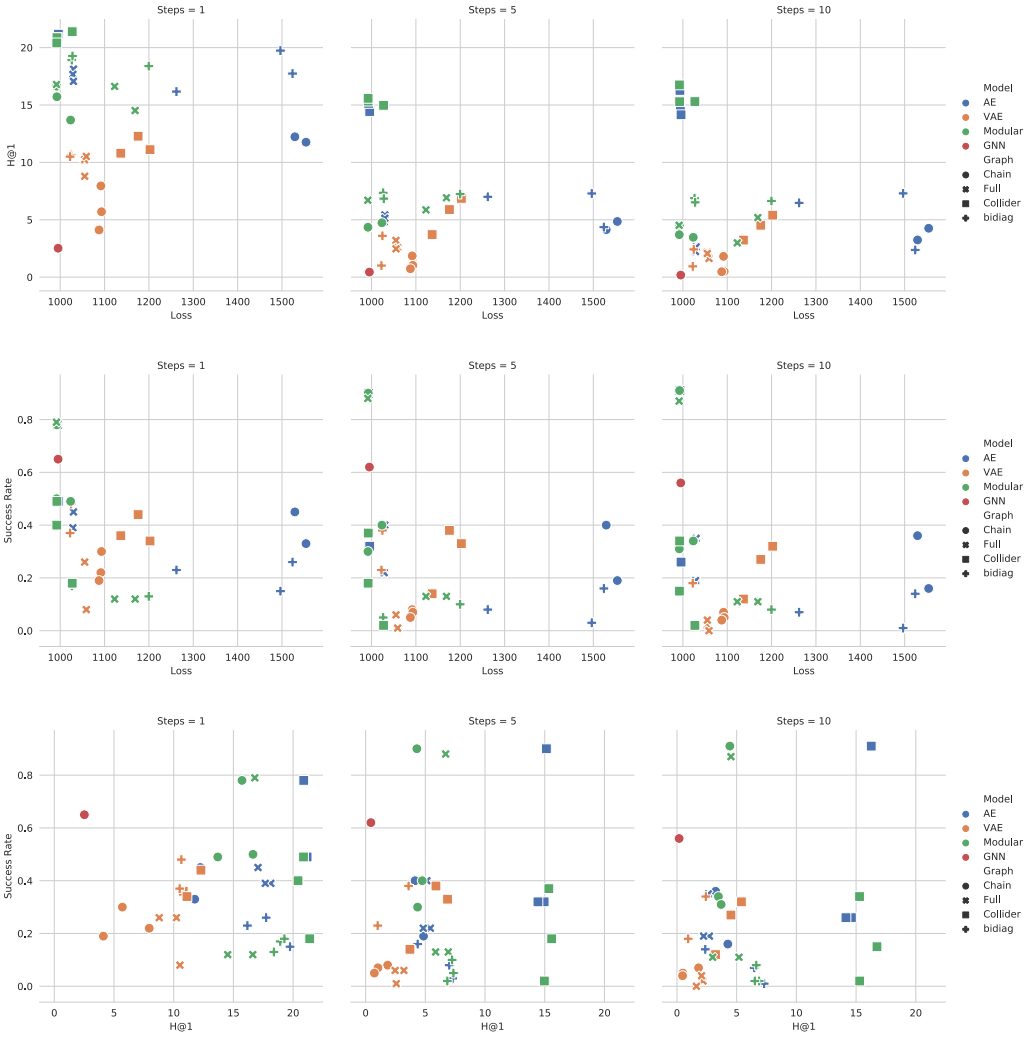

Figure 28: Plots for chemistry environment with 5 objects and 5 colors for models trained using NLL Loss. We see that there seems to be a positive correlation between H@1 and success rate for step 1 but this may not be true for longer steps.