
Compete and Compose: Learning Independent Mechanisms for Modular World Models

Anson Lei
Applied AI Lab
University of Oxford, UK
anson@robots.ox.ac.uk

Frederik Nolte
Applied AI Lab
University of Oxford, UK
frederiknolte@robots.ox.ac.uk

Bernhard Schölkopf
MPI for Intelligent Systems
Tübingen, Germany
bs@tue.mpg.de

Ingmar Posner
Applied AI Lab
University of Oxford, UK
ingmar@robots.ox.ac.uk

Abstract

We present *Competitive Mechanisms for Efficient Transfer* (COMET), a modular world model which leverages reusable, independent mechanisms across different environments. COMET is trained on multiple environments with varying dynamics via a two-step process: competition and composition. This enables the model to recognise and learn transferable mechanisms. Specifically, in the competition phase, COMET is trained with a winner-takes-all gradient allocation, encouraging the emergence of independent mechanisms. These are then re-used in the composition phase, where COMET learns to re-compose learnt mechanisms in ways that capture the dynamics of intervened environments. In so doing, COMET explicitly reuses prior knowledge, enabling efficient and interpretable adaptation. We evaluate COMET on environments with image-based observations. In contrast to competitive baselines, we demonstrate that COMET captures recognisable mechanisms without supervision. Moreover, we show that COMET is able to adapt to new environments with varying numbers of objects with improved sample efficiency compared to more conventional finetuning approaches.

1 Introduction

To reason about environments as rich and complex as our physical world requires the ability to learn efficiently and to flexibly adapt prior knowledge to unseen settings. Whilst humans seem able to generalise knowledge across myriad tasks and situations effortlessly, building artificial agents that can do so with minimal training data remains a significant challenge. At the heart of this challenge are crucial distinctions between what and how humans and machines learn. It has been conjectured that humans represent knowledge internally in a structured and modular way, i.e., by distilling past experience into general principles (or core knowledge) about the world, which can be applied or selectively updated in novel settings [43, 26, 39, 11]. By contrast, current learning-based world models are mostly based on monolithic architectures, and the resulting entangled representations of the world limit the selective re-using of prior knowledge in new environments. Therefore,

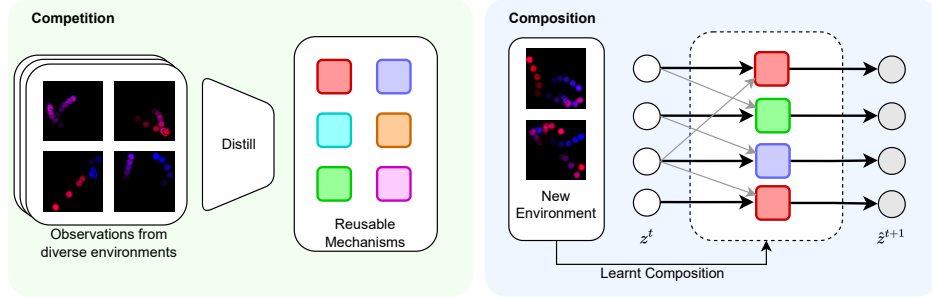


Figure 1: Illustration of the *competition* and *composition* training phases. In the first phase, the model learns a set of reusable mechanisms that captures interaction primitives. In the second phase, the model learns to apply these mechanisms in a new environment.

learning methods that afford modularity are key to world models that can adapt efficiently in diverse settings. In this paper, we address this challenge by developing a model capable of discovering a toolbox of recognisable, generalisable concepts that can be reused across different contexts.

Recent works on object-centric world models [24, 28] serve as an initial step towards a structured and compositional understanding of the world. By decomposing the observed scene into discrete *object slots*, these methods model the interaction between entities in the scene and achieve state-of-the-art results. We argue that, just as the state representation of the scene can be factorised into object slots, the dynamics of the environment, too, can be factorised into discrete and independent *mechanisms*. Our work is motivated by the observation that, while the overall dynamics can change across environments, we can often explain the behaviour of objects by a small set of *interaction primitives* such as "A rests on top of B" and "C collides with D". Our understanding of novel environments is mediated by these primitives: given some observations in a previously unseen setting, one can quickly recognise the relationship between objects in terms of these interaction primitives. As such, we posit that the ability to structurally represent different modes of interactions is crucial to flexible world models.

Acquiring such a set of versatile mechanisms from observations without supervision presents challenges both in terms of model architecture and learning algorithm. While current learning methods excel at learning to predict dynamics from i.i.d. environments, learning shared modules that compositionally capture diverse environments cannot be achieved by directly applying gradient updates to the entire model. To this end, we argue that the ability to *selectively* update the model during learning, i.e., to recognise parts of the model that are relevant to the observed data and perform modular updates, is instrumental to the emergence of discrete independent mechanisms. In this work, we instantiate this capability by utilising a competition of experts training scheme [36], where, during training, we keep a set of independently parameterised mechanisms and only update the module which best explains the observed data. This serves as a natural inductive bias which encourages modules to specialise in specific interaction primitives, and is consequently conducive to the emergence of reusable mechanisms.

We introduce *Competitive Mechanisms for Efficient Transfer* (COMET), a modular world model with the capability to explicitly disentangle modes of interactions between objects. In contrast with prior art, COMET is able to learn discrete, abstract concepts from diverse observations, and re-use such concepts to predict the evolution of unseen environments. We conjecture that COMET’s ability to perceive the world through the lens of structured high-level abstract concepts serves as an important step towards efficient generalisation and transfer across different task settings. Concretely, COMET achieves this via a two-phase training procedure: i) *competition*, in which COMET learns a set of independently parameterised modules which encode interaction primitives from diverse environments; and ii) *composition*, wherein COMET is trained to apply these learnt interaction primitives in novel environments (Fig. 1).

In our experiments, we evaluate COMET on novel datasets designed to test whether mod-

els can systematically learn from multiple environment and transfer to environments with unseen dynamics. We demonstrate both quantitatively and qualitatively that COMET successfully learns reusable mechanisms, i.e. interaction primitives, from image-based observations, and is able to generalise to unseen environments by composing these mechanisms. Moreover, compared to more conventional finetuning baselines, COMET is able to explicitly make use of past knowledge and achieves improved sample efficiency during adaptation.

2 Related Work

Learning internal models of the world enables decision-making agents to plan, predict and reason about the world [17, 16, 24]. As such, latent world models have attracted significant interest in recent years. These methods [e.g. 24, 48, 34] in general involve learning latent representations of the state and forward prediction models. Our work situates in this broad context of world model learning and we focus our contribution on learning dynamics models which are factorised into composable mechanisms. We take recent works from object-centric state representations as a starting point.

Object-Centric Representations There has been a growing interest in models that reflect the compositional nature of real-world scenarios and aim to use object-centric representations to leverage recurring features in scenes. Prior works have investigated unsupervised object-centric representation learning from static images [5, 15, 32, 29, 10]. Motivated by the assumption that dynamics tend to manifest themselves at the object-level [18, 3], subsequent works extend this capability to video data via factorised dynamics models which operate on object-centric latent spaces. While most of these object-centric world models (OCWMs) are geared towards using temporal inputs to generate future video rollouts [25, 21, 8, 24, 28, 33], some more explicitly consider their use in model-based reinforcement learning and planning [50, 47, 41, 51]. In particular, graph neural networks (GNNs) are often used as a natural way to predict future states of objects and enable the modelling of interactions between objects via message passing [45, 44, 24, 47, 28, 41, 37]. We build on these approaches by further factorising the dynamics into reusable interaction primitives.

Mechanism-based Models Our work is motivated by the conjecture that the organisation of knowledge into high-level abstract concepts is crucial to systematic generalisation [11]. This idea is similar in spirit to the *Independent Causal Mechanisms* principle [38] and the *Sparse Mechanism Shift* hypothesis [39] in the causality literature, which respectively posit that data-generating causal mechanisms operate independently from one another, and that changes in the environment can be attributed to sparse changes to such mechanisms. Several works [27, 19, 30] have leveraged causal discovery techniques, e.g., sparsity regularisation, to learn dynamics models that are factorised into structural causal models.

Similar to our approach is a class of models which represents the learned dynamics in OCWMs not as a monolithic module, but rather as a collection of independently acting mechanisms – each focusing on a different aspect of the environment’s dynamics. Becker-Ehmck et al. [4] use a variational approach to learn to pick different transition models conditioned on the state, but is limited to linear transitions. RIMs [14] constitute an approach where parts of the state space are represented by independent and sparsely interacting recurrent units. Building on this, [12] use a GNN to model environment dynamics but reflect the concept of independent mechanisms by using different sets of GNN parameters depending on an object’s current state. Another approach that follows this line of work is VIM [2] which considers the disentanglement of mechanisms and objects in the setting where object move independently to each other. Closer to our method are Neural Production Systems (NPS) [13], another descendant of RIMs, learning a set of independent mechanisms capturing the interaction between objects. Our method differs from NPS in the application of competition training which, as we demonstrate empirically in Sec. 4, is instrumental to the emergence of composable mechanisms. Furthermore, we propose a novel method for adapting to changes in the environment.

Competition of Experts The backbone of our learning algorithm draws from mixture of experts methods [22, 20, 40] and in particular from the algorithm of Parascandolo et al. [36]. In the context of learning independent causal mechanisms, Parascandolo et al. [36] demonstrate that the competition of experts algorithm induces the emergence of mechanisms that explain transformations in the data. The idea of utilising a competitive training scheme on modular model architectures has been applied on diverse settings such as lifelong learning [1, 35], generative models [31] and object-centric scene

composition [49]. Taking inspiration from this line of work, COMET uses a similar competitive training scheme as an inductive bias for disentangling modes of interaction in the setting of world model learning.

3 COMET: Competitive Mechanisms for Efficient Transfer

In this section, we present the training procedure and the architecture of COMET. The main idea of the method is to learn a set of generalisable and composable modules, which encode the different modes of interaction between objects. The intuition behind the approach is that, while dynamics can vary across environments, the ways in which objects or entities interact with each other can often be explained by a small number of independent rules. For example, in a road traffic setting, whilst the behaviour of cars can differ across different locations, the act of stopping at a red light, i.e. the interaction between cars and traffic lights, can be used to explain road behaviours in a wide range of environments. On a conceptual level, this can be considered as a manifestation of the ICM principle [39]. By having a model of *how* objects could interact with each other, the task of adapting to a novel environment reduces to the learning of *when* each rule should be applied.

The goal of COMET is to discover potential interactions between objects through observed data from a diverse set of environments, and to adapt to novel environments by learning when to utilise learnt mechanisms. This is reflected in the training strategy of COMET, which is split into two phases: *competition* and *composition*. In the *competition* phase, COMET learns a set of independent mechanisms from observed sequences using a competition of experts training scheme. In the *composition* phase, COMET adapts to novel environments by learning a classifier that activates the correct mechanism with the correct object pair based on the state. Our main hypothesis is that learning to apply pre-trained, reusable mechanisms in a novel environment facilitates explicit transfer of past knowledge and is hence more efficient than finetuning on new data.

3.1 Problem Setup and Model Architecture

COMET learns from a dataset of observed sequences $\{\mathbf{x}^{1:T}\}_N$. In this work we focus on learning from observations without actions, which is commensurate with the settings in similar works [28, 21], although the framework presented here can be readily extended to action-conditioned world models. Importantly, these sequences are sampled from environments with varying dynamics where objects can exhibit different behaviours. In order to model interactions between objects, we assume that each observation, \mathbf{x}^t , can be factorised into latent object-slots, $\{\mathbf{z}_0^t, \mathbf{z}_1^t, \dots, \mathbf{z}_K^t\}$, where the subscript denotes the object-id. These object representations can be based on ground-truth state information or obtained from object-centric encoders [5, 32, 10].

COMET consists of two main components, the *mechanisms* and the *composition module*. *Mechanisms* contains M independently parameterised feedforward networks, $f_{mech}^m : \mathbb{R}^{2d} \rightarrow \mathbb{R}^d$, with parameters θ_m , where d is the dimension of the object representations. Each mechanism predicts updates to all objects at every timestep, given the state of the object itself and another context object:

$$\Delta \mathbf{z}_i^t(m, j) = f_{mech}^m([\mathbf{z}_i^t \oplus \mathbf{z}_j^t]), \quad (1)$$

where \oplus denotes concatenation, i is the index of the object to be predicted and j is the index of the *context* object, i.e. the object with which object i interacts. The mechanisms are trained during the *competition* phase where each mechanism learns to specialise to cover a particular mode of interaction between objects.

In order to predict transitions for objects using the trained mechanisms, the composition module picks the relevant context object and the active mechanism. The composition module predicts the mechanism-context pair based on the observed state:

$$(\hat{m}_i, \hat{j}_i) = f_{comp}(\mathbf{z}_i^t, \mathbf{z}_{1:K}^t), \quad (2)$$

where (\hat{m}_i, \hat{j}_i) is the predicted mechanism-context pair for object i and $\mathbf{z}_{1:K}$ is the representation of all objects. f_{comp} can be considered as a classifier with $M \times K$ classes, where M is the number of mechanisms and K is the number of objects in the scene. In the *composition* phase, COMET adapts to a new environment given a small number of observed sequences $\{\tilde{\mathbf{x}}^{1:T}\}_{\tilde{N}}$ by updating the composition module.

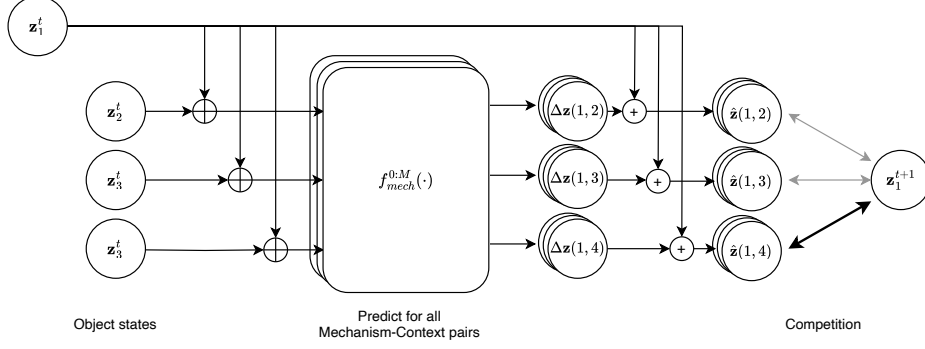


Figure 2: In the *competition* phase, predictions are made using all possible mechanism-context pairs for each object. Gradients are only allocated to the mechanism-context pair which produces the most accurate prediction. This encourages specialisation within the mechanisms and enables learning from environments with varying dynamics. The figure describes the prediction step for a single object.

3.2 Phase 1: Learning Reusable Mechanisms via Competition

The aim of the *competition* phase is to train the mechanisms in a way such that each mechanism specialises in a particular mode of interaction between objects. Since the training dataset is sampled from multiple environments with varying dynamics, the training procedure needs to recognise shared mechanisms regardless of context. Taking inspiration from Parascandolo et al. [36], we apply a competitive training scheme in the setting of dynamics learning. Concretely, for each object, the model makes predictions using all possible mechanism-context pairs in parallel. Comparing the predictions, we update the mechanism-context pair with the most accurate prediction for a given object. This is illustrated in Fig. 2. Given a state transition pair, $(z_{1:K}^t, z_{1:K}^{t+1})$, the loss function can be written as:

$$\mathcal{L}(\theta_{1:M}) = \sum_{i=0}^K \min_{m,j} [d(z_i^t + \Delta z_i^t(m, j), z_i^{t+1})], \quad (3)$$

where d is a function that measures the prediction error. There are several sensible choices for the distance function d . In this work, we find that using Euclidean distance is sufficient. Other potential choices include a contrastive loss [24] or ELBO loss [17] for training representation and dynamics model jointly, and sparsity-regularised loss [27, 19] for causal structure discovery. Importantly, when performing back-propagation on this loss function, only the parameters of the competition winner are updated. This independent update of parameters encourages *specialisation* among the mechanisms and facilitates environment-agnostic learning. Intuitively, insofar as a particular mode of interaction is concerned, a specialised mechanism will outperform other mechanisms and hence win more data as training progresses. This triggers a positive feedback loop which allows the specialised mechanism to further improve its accuracy. Moreover, the competition procedure ensures the selection of the correct context objects during training since accurate predictions are only possible with the relevant context information.

In practice, one of the main failure modes in this training phase is a case where a single mechanism wins the competition most of the time, which hinders the proper disentanglement of dynamics. Empirically, we find that the robustness of the training dynamics can be significantly improved by 1) adding a warm-starting phase during which gradients are equally distributed; and by 2) increasing the time horizon for mechanism selection which improves the robustness of the training dynamics. This means that a mechanism-context pair is picked only if it has the best performance over multiple consecutive time steps. In App. C, we discuss this further with ablation studies.

3.3 Phase 2: Learning to Compose Mechanisms in New Environments

With a trained set of mechanisms, COMET adapts to new environments by training the composition module on new data.¹ The composition module acts as a classifier which picks the optimal mechanism-context pair for each object in the scene. Given the state of object i , for each mechanism-context pair, we compute the confidence score:

$$c_i^t(m, j) = f_{conf}^m([z_i^t \oplus z_j^t]), \quad (4)$$

where f_{conf}^m is an independently parameterised MLP for mechanism m , i.e. each mechanism has a corresponding f_{conf}^m . We predict a categorical distribution over all mechanism-context pairs by taking the softmax over the confidence scores for object i at time step t .

Given a small number of observation sequences in a new environment, we obtain the best performing mechanism-context pair $(m^*, j^*)_i^t$ for each object at each time step by investigating which pair minimises the loss function of the competition scheme in Eq. 3. These then serve as the target labels for the classifier. The composition module is then trained using the negative log-likelihood loss. At test time, together with the trained mechanisms, COMET predicts the next state of each object by first picking the mechanism-context pair with the highest score and then feeding the chosen context object through the chosen mechanism. In the following section, we show empirically that this two-step training procedure is able to learn meaningful mechanisms from image observations and can compose mechanisms in novel environments efficiently.

4 Experiments

In this section, we demonstrate that COMET is able to disentangle different modes of interaction between objects and can efficiently reuse learnt mechanisms during adaptation. We evaluate, both quantitatively and qualitatively, the performance of COMET on image-based environments. Concretely, the experiments focus on whether COMET can learn reusable and recognisable mechanisms (Sec. 4.2) and whether re-composing learnt mechanisms is more sample-efficient compared to finetuning approaches (Sec. 4.3).

4.1 Experimental Setup

Baselines. We evaluate COMET against two competitive baselines, C-SWM [24] and Neural Production Systems (NPS) [13]. C-SWM learns a world model from observation via contrastive learning with a GNN-based dynamics model. Similar to COMET, C-SWM operates on an object-factorised representation and achieves state-of-the-art prediction accuracy. COMET further disentangles the interactions between objects as independent mechanisms rather than learning a monolithic model that captures all interactions. In Sec. 4.3, we show that this disentanglement is conducive to sample-efficient adaptation and facilitates interpretable transfer.

We also compare against NPS, which learns to capture object interactions as independent mechanisms. Architecturally, NPS is similar to COMET, except that the mechanism-context pair is selected using dot-product attention [46] which is trained jointly with the mechanisms. In contrast, COMET deploys a competitive training scheme which allows the model to recognise shared mechanisms across environments. Moreover, in Sec. 4.2, we show that competition serves as a strong inductive bias that enables the emergence of generalisable mechanisms. We provide more model details and further discussions around the baselines in App. A.

Datasets. We evaluate COMET on three problem domains: *Particle Interactions*, *Traffic*, and *Team Sports*. For each of these domains, we define a set of environments where objects can exhibit different behaviours. These environments are designed to test whether COMET can extract meaningful mechanisms and adapt to unseen environments via composition. The *Particle Interactions* dataset consists of coloured particles that can interact with each other in different ways such as attraction and repulsion. Environments are defined by a combination of rules such as "red particles repel each other". The *Traffic* dataset contains observation sequences of traffic scenarios generated with the CARLA

¹Here, we focus on directly reusing mechanisms new environments. We leave the exploration of systems that can instantiate new mechanisms to future work. See discussion in Sec. 4.4.

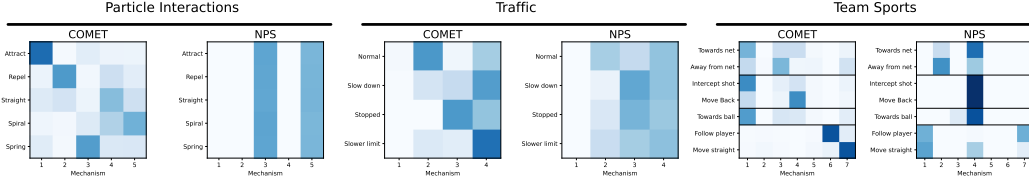


Figure 3: Disentanglement plots showing the correlation between mechanisms chosen by the models and ground-truth interaction modes. In the ideal case, the matrices should look like permutation matrices. Here, COMET is able to learn disentangled mechanisms that correspond to ground-truth behaviours in all three domains, as indicated by the fact that each interaction mode has one main corresponding learnt mechanism. In contrast, NPS does not exhibit the same structure.

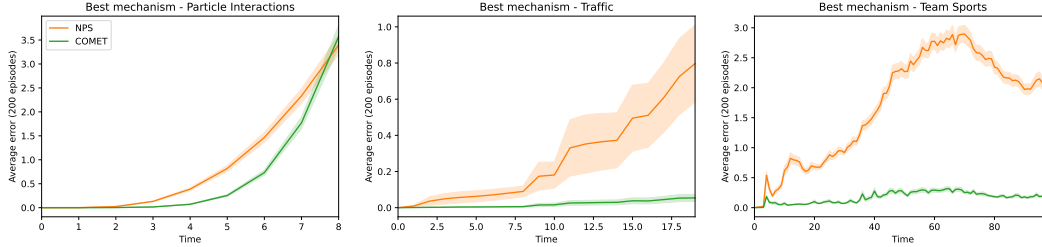


Figure 4: Rollout errors (lower is better) in unseen environments with optimal mechanism selection. Shaded areas indicates the standard error of the mean. The lower errors indicate that COMET mechanisms can be readily reused across environments *without* finetuning.

simulator [6]. Here, the environments are defined by traffic rules that apply to different vehicles such as "blue cars do not need to stop at red lights". The *Team Sport* domain consists of a simulated generic hockey game where players can perform different actions such as moving towards the puck or dribbling the puck towards the opponent goal, generated by the STS2 simulator [42]. In this setting, differences in environment amounts to the different behaviours of the players, e.g. players in some environments might tend to take more aggressive actions. This is a particularly challenging dataset for COMET as it violates the assumption that all interactions are binary. Observations are masked RGB images for each object, except for the *Team Sports* domain, where state-based observations are more naturally suited. Details of the datasets are provided in App. B.

4.2 Disentanglement of Mechanisms

We investigate the emergence of recognisable mechanisms from competition. Here, both COMET and NPS are trained on a mixture of environments in each domain. We obtain the ground-truth labels for the object interactions and use these to directly investigate whether the learnt mechanisms correspond to actual interactions without any supervision. These labels are not accessible to the models during training. Fig. 3 shows the correlation between the active ground-truth interactions and the winning mechanisms in the competition process in the different domains. COMET achieves successful disentanglement and learns mechanisms that corresponds to the ground-truth interactions. In the *particle interaction* dataset, COMET recovers the ground-truth mode of interactions between particles. Similarly, in the *Traffic* and *Team Sports* datasets, COMET learns mechanisms that model interaction primitives such as stopping before a red light and running towards the opponent goal.² Fig. 5 qualitatively illustrate that the 'winning' mechanisms switches as the underlying interaction type changes. In contrast, the mechanisms learnt by NPS show no correspondence with the ground-truth interactions. This is likely because NPS cannot learn from a mixture of environments with varying dynamics as it employs a simple dot-product attention for picking

²In the team sports dataset, we observe that mechanism 1 is 'overloaded' in the sense that it represents different behaviours for different objects, i.e. for the attacking team, it models 'move towards net' and for the defending team, it models 'intercept shot'. Crucially, for each object type (indicated by the horizontal divider lines in Fig. 3), different behaviours are captured with different mechanisms. This is not the case for NPS.

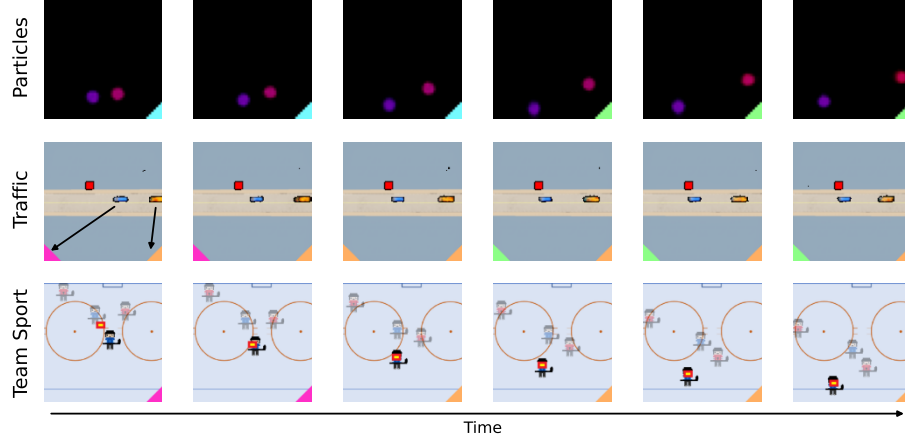


Figure 5: Qualitative rollouts. The colour of the tabs on the bottom of each frame indicates the ‘winning’ mechanism at each time step. Across all environments, the competition winner changes as the underlying interaction mode changes. **Top:** The particles repel each other when they are close (blue) and moves independently when they are apart (green). **Middle:** In this traffic environment, the orange car obeys a slower speed limit and always pick the slow mechanism (orange). The blue car approaches the red light with normal driving (pink) → slow down (orange) → stop (green). Note that the orange mechanism is used as slow driving for both cars. **Bottom:** The player first wait to receive the ball (pink) and the moves towards opponent goal when in pocession of the ball (orange).

mechanisms during training. To this end, COMET’s ability to learn from diverse environments is uniquely afforded by the competition scheme which assigns relevant data to update each mechanism.

Central to our hypothesis is the notion that good mechanisms are *reusable* across environments. To this end, we perform rollout in *unseen* environments by picking the optimal mechanism-object pair, (m^*, j^*) , effectively bypassing the confidence module. This offers insights to whether the learnt mechanisms can generalise to new settings. We perform this to the mechanisms learnt with COMET and NPS. Fig. 4 shows the rollout errors with optimal mechanism selection. COMET mechanisms performs significantly better than NPS mechanisms, meaning that, when picked correctly, COMET mechanisms can be directly used in new environment without any finetuning. This corroborates our claim that competition training is conducive to the emergence of generalisable and reusable mechanisms.

4.3 Adaptation Efficiency

One of our main hypotheses is that learning to compose learnt mechanisms leads to data-efficient adaptation. For each domain, we train all of the models on a mixture of environments and adapt the models to unseen environments. COMET adapts by training the composition module, whereas the baselines adapt by finetuning the entire model on new data. Fig. 6 shows the prediction performance of the models when trained on different amounts of observations in the new environment. In general, all models improve as the amount of available data increases. In the *Particle Interactions* and *Traffic* domains, while all models eventually perform comparably given sufficient data, here we see that COMET outperforms the baselines in the low-data regime, illustrating that explicitly reusing learnt mechanisms results in improved sample efficiency compared to gradient-based finetuning. In the *Team Sport* domain, C-SWM performs better than COMET and NPS. We hypothesise that this is because both COMET and NPS only model binary interactions between objects whereas C-SWM uses a GNN-based transition model that can take into account the entire scene. We further discuss this limitation in Sec. 4.4. Nonetheless, under the same binary interactions restriction, COMET significantly outperforms NPS, which overfits to the adaptation data quickly and is not able to generate stable rollouts.

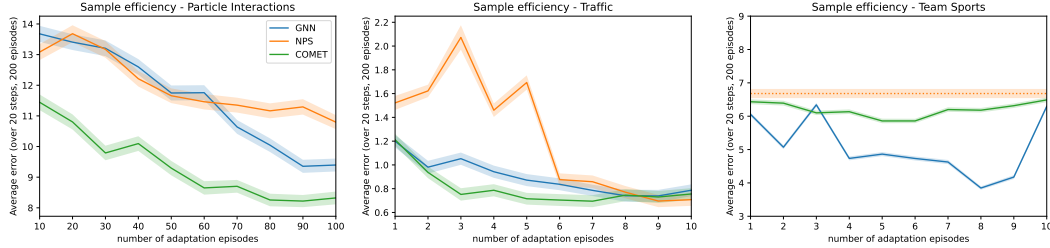


Figure 6: The average rollout error in an unseen environment with different amount of observed data in the new environment (lower is better). In all environments, all models eventually converge to similar errors given enough data. We show this explicitly in App D and offer further discussion. In terms of sample efficiency, in the *Particle Interactions* and *Traffic* domains, COMET is able to achieve lower errors with few adaptation episodes. This means that COMET can learn to use the correct mechanisms with a small amount of data, thus corroborates our hypothesis that composing learnt mechanisms enables sample-efficient transfer. In the *Team Sports* domain, NPS is not able to generate stable rollouts with the amounts of adaptation episodes shown in the plots. The dotted line indicates the performance of NPS when trained with a large amount of data. Shaded areas represent the standard errors of the mean.

4.4 Limitations

Object Encoders. In our experiments, we focus our analysis on the dynamics models and opt for simple CNN encoders with ground-truth object segmentation masks as the representation backbone. The application of the model on more visually complex environments without segmentation masks would require the use of object-centric encoders or other segmentation techniques [7, 9, 32].

Binary Interactions. The model architecture for the mechanisms implies that the model can only capture *binary* interactions. While this is a good approximation for many interacting systems, this hinders the model’s ability to capture more complex dynamics. As seen in the Team Sports experiments, while COMET can disentangle meaningful mechanisms, picking correct mechanisms may require information from the entire scene, leading to suboptimal prediction accuracy. Considering n -ary interactions naively would require comparing exponentially many options in the competition phase. We leave the exploration of higher-order interactions to future work.

Adapting Mechanisms. As it stands, COMET lacks the ability to update the mechanisms during *composition*, which limits the model’s ability to adapt to environments with completely new interactions. Designing mechanism-based models that can instantiate new mechanisms as it encounters new data offers an exciting avenue of research, with important implications on life-long learning, where an agent can improve its understanding of the world through a growing set of reusable mechanisms. This is beyond the scope of the present investigation. Nonetheless, we believe the method presented here will be instrumental in the development of such systems.

5 Conclusion

In this paper, we introduce COMET, a structured world model which encodes discrete abstract mechanisms explicitly from observations. This is achieved through training the model on a set of environments with diverse dynamics using a winner-takes-all competition scheme. This enables the model to perform *selective* updates during the training phase, a central capability which facilitates the emergence of recognisable and reusable mechanisms. We show experimentally that the proposed method is indeed able to disentangle shared mechanisms across different environments from image observations, and thus enables sample-efficient and interpretable adaptation to novel situations. While the presented results are competitive on the tasks considered, we see our main contribution as a conceptual one, serving as a step towards world models exhibiting a structured understanding of the world. Looking forward, we believe that the method developed here opens up several promising avenues of research, such as designing agents that learn a growing repertoire of re-usable interaction behaviours and agents that explore the world through the lens of mechanism discovery.

Acknowledgements

This research was supported by an EPSRC Programme Grant (EP/V000748/1). The authors would like to acknowledge the use of the SCAN facility

References

- [1] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3366–3375, 2017.
- [2] Rim Assouel, Lluís Castrejon, Aaron Courville, Nicolas Ballas, and Yoshua Bengio. VIM: Variational independent modules for video prediction. In Bernhard Schölkopf, Caroline Uhler, and Kun Zhang, editors, *Proceedings of the First Conference on Causal Learning and Reasoning*, volume 177 of *Proceedings of Machine Learning Research*, pages 70–89. PMLR, 11–13 Apr 2022. URL <https://proceedings.mlr.press/v177/assouel122a.html>.
- [3] David Baraff. Physically based modeling: Rigid body simulation. *SIGGRAPH Course Notes, ACM SIGGRAPH*, 2(1):2–1, 2001.
- [4] Philip Becker-Ehmck, Jan Peters, and Patrick Van Der Smagt. Switching linear dynamics for variational bayes filtering. In *International conference on machine learning*, pages 553–562. PMLR, 2019.
- [5] Christopher P. Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. MONet: Unsupervised Scene Decomposition and Representation. *arXiv preprint arXiv:1901.11390*, 2019. doi: 10.48550/ARXIV.1901.11390.
- [6] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.
- [7] Gamaleldin F. Elsayed, Aravindh Mahendran, Sjoerd van Steenkiste, Klaus Greff, Michael C. Mozer, and Thomas Kipf. SAVi++: Towards end-to-end object-centric learning from real-world videos. In *Advances in Neural Information Processing Systems*, 2022.
- [8] Patrick Emami, Pan He, Anand Rangarajan, and Sanjay Ranka. A Symmetric and Object-Centric World Model for Stochastic Environments. In *NeurIPS 2020 Workshop on Object Representations for Learning and Reasoning*, 2020.
- [9] M. Engelcke, O. Parker Jones, and I. Posner. Genesis-v2: Inferring unordered object representations without iterative refinement. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8085–8094. Curran Associates, Inc., 2021.
- [10] Martin Engelcke, Oivi Parker Jones, and Ingmar Posner. GENESIS-V2: Inferring Unordered Object Representations without Iterative Refinement. In *Advances in Neural Information Processing Systems*, volume 34, pages 8085–8094. Curran Associates, Inc., 2021.
- [11] Anirudh Goyal and Yoshua Bengio. Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 478(2266):20210068, 2022. doi: 10.1098/rspa.2021.0068.
- [12] Anirudh Goyal, Alex Lamb, Phanideep Gampa, Philippe Beaudoin, Sergey Levine, Charles Blundell, Yoshua Bengio, and Michael Mozer. Object Files and Schemata: Factorizing Declarative and Procedural Knowledge in Dynamical Systems. *arXiv preprint arXiv:2006.16225*, November 2020. doi: 10.48550/arXiv.2006.16225.
- [13] Anirudh Goyal, Aniket Rajiv Didolkar, Nan Rosemary Ke, Charles Blundell, Philippe Beaudoin, Nicolas Heess, Michael Curtis Mozer, and Yoshua Bengio. Neural production systems. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.
- [14] Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent Independent Mechanisms. In *International Conference on Learning Representations*, January 2021.

- [15] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-Object Representation Learning with Iterative Variational Inference. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2424–2433. PMLR, May 2019.
- [16] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems 31*, pages 2451–2463. Curran Associates, Inc., 2018.
- [17] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning Latent Dynamics for Planning from Pixels. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2555–2565. PMLR, May 2019.
- [18] Patrick J Hayes. The naive physics manifesto. *Expert systems in the microelectronic age*, 1979.
- [19] Biwei Huang, Fan Feng, Chaochao Lu, Sara Magliacane, and Kun Zhang. AdaRL: What, where, and how to adapt in transfer reinforcement learning. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=8H5bpVwvt5>.
- [20] Robert Jacobs and Michael Jordan. A competitive modular connectionist architecture. *Advances in neural information processing systems*, 3, 1990.
- [21] Jindong Jiang, Sepehr Janghorbani, Gerard De Melo, and Sungjin Ahn. SCALOR: Generative World Models with Scalable Object Representations. In *International Conference on Learning Representations*, 2020.
- [22] Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.
- [23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [24] Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. In *International Conference on Learning Representations*, 2020.
- [25] Adam Kosior, Hyunjik Kim, Yee Whye Teh, and Ingmar Posner. Sequential Attend, Infer, Repeat: Generative Modelling of Moving Objects. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.
- [26] Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253, 2017. doi: 10.1017/S0140525X16001837.
- [27] Anson Lei, Bernhard Schölkopf, and Ingmar Posner. Variational causal dynamics: Discovering modular world models from interventions. *CoRR*, abs/2206.11131, 2022. doi: 10.48550/arXiv.2206.11131. URL <https://doi.org/10.48550/arXiv.2206.11131>.
- [28] Zhixuan Lin, Yi Fu Wu, Skand Peri, Bofeng Fu, Jindong Jiang, and Sungjin Ahn. Improving generative imagination in object-centric world models. In *37th International Conference on Machine Learning, ICML 2020*, volume Part F16814, pages 6096–6105, 2020. ISBN 978-1-71382-112-0.
- [29] Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. SPACE: Unsupervised Object-Oriented Scene Representation via Spatial Attention and Decomposition. In *International Conference on Learning Representations*, 2020.
- [30] Phillip Lippe, Sara Magliacane, Sindy Löwe, Yuki M Asano, Taco Cohen, and Stratis Gavves. Citris: Causal identifiability from temporal intervened sequences. In *International Conference on Machine Learning*, pages 13557–13603. PMLR, 2022.
- [31] Francesco Locatello, Damien Vincent, Ilya Tolstikhin, Gunnar Rätsch, Sylvain Gelly, and Bernhard Schölkopf. Competitive training of mixtures of independent deep generative models. *arXiv preprint arXiv:1804.11130*, 2018.
- [32] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-Centric Learning with Slot Attention. In *Advances in Neural Information Processing Systems*, volume 33, pages 11525–11538. Curran Associates, Inc., 2020.
- [33] Cheol-Hui Min, Jinseok Bae, Junho Lee, and Young Min Kim. GATSBI: Generative Agent-Centric Spatio-Temporal Object Interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3074–3083, 2021.

- [34] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- [35] Oleksiy Ostapenko, Pau Rodriguez, Massimo Caccia, and Laurent Charlin. Continual learning via local module composition. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL <https://openreview.net/forum?id=LJjC6DmSkGT>.
- [36] Giambattista Parascandolo, Niki Kilbertus, Mateo Rojas-Carulla, and Bernhard Schölkopf. Learning independent causal mechanisms. In *International Conference on Machine Learning*, pages 4036–4044. PMLR, 2018.
- [37] Cansu Sancaktar, Sebastian Blaes, and Georg Martius. Curious Exploration via Structured World Models Yields Zero-Shot Object Manipulation. In *Advances in Neural Information Processing Systems*, October 2022.
- [38] Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On causal and anticausal learning. In *Proceedings of the 29th International Conference on Machine Learning*, pages 459–466, 2012.
- [39] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Towards causal representation learning. *Proceedings of the IEEE*, 109(5): 612–634, May 2021. ISSN 1558-2256. doi: 10.1109/JPROC.2021.3058954.
- [40] Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL <https://openreview.net/forum?id=B1ckMDqlg>.
- [41] Gautam Singh, Skand Peri, Junghyun Kim, Hyunseok Kim, and Sungjin Ahn. Structured World Belief for Reinforcement Learning in POMDP. In *Proceedings of the 38th International Conference on Machine Learning*, pages 9744–9755. PMLR, July 2021.
- [42] Caedmon Somers, Jason Rupert, Yunqi Zhao, Igor Borovikov, Jiachen Yang, and Ahmad Beirami. Simple Team Sports Simulator (STS2), February 2020.
- [43] Elizabeth S Spelke and Katherine D Kinzler. Core knowledge. *Developmental science*, 10(1):89–96, 2007.
- [44] Andrea Tacchetti, H. Francis Song, Pedro A. M. Mediano, Vinicius Zambaldi, János Kramár, Neil C. Rabinowitz, Thore Graepel, Matthew Botvinick, and Peter W. Battaglia. Relational Forward Models for Multi-Agent Learning. In *International Conference on Learning Representations*, December 2018.
- [45] Sjoerd van Steenkiste, Michael Chang, Klaus Greff, and Jürgen Schmidhuber. Relational Neural Expectation Maximization: Unsupervised Discovery of Objects and their Interactions. In *International Conference on Learning Representations*, 2018.
- [46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [47] Rishi Veerapaneni, John D. Co-Reyes, Michael Chang, Michael Janner, Chelsea Finn, Jiajun Wu, Joshua Tenenbaum, and Sergey Levine. Entity Abstraction in Visual Model-Based Reinforcement Learning. In *Proceedings of the Conference on Robot Learning*, pages 1439–1456. PMLR, May 2020.
- [48] Ruben Villegas, Arkanath Pathak, Harini Kannan, Dumitru Erhan, Quoc V Le, and Honglak Lee. High fidelity video prediction with large stochastic recurrent neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.
- [49] Julius von Kügelgen, Ivan Ustyuzhaninov, Peter Gehler, Matthias Bethge, and Bernhard Schölkopf. Towards causal generative scene models via competition of experts. *arXiv preprint arXiv:2004.12906*, 2020.
- [50] Nicholas Watters, Loic Matthey, Matko Bosnjak, Christopher P. Burgess, and Alexander Lerchner. COBRA: Data-Efficient Model-Based RL through Unsupervised Object Discovery and Curiosity-Driven Exploration. *arXiv preprint arXiv:1905.09275*, August 2019. doi: 10.48550/arXiv.1905.09275.
- [51] Yizhe Wu, Oiwi Parker Jones, Martin Engelcke, and Ingmar Posner. APEX: Unsupervised, Object-Centric Scene Segmentation and Tracking for Robot Manipulation. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3375–3382, May 2021.

	Layer Type	in Features	out Features
1	Linear	2 * 8	300
2	ELU		
3	Linear	300	300
4	ELU		
5	Linear	300	8

Table 1: Architecture of a single mechanism. The mechanism operates on the concatenation of two object embeddings and predicts state updates for the first input object.

	Layer Type	in Features	out Features
1	Linear	2 * 8	300
2	ELU		
3	Linear	300	300
4	ELU		
5	Linear	300	1

Table 2: Architecture of a single confidence network. There is one confidence network per mechanism and it operates on the concatenation of two object embeddings to predict a score for the given context object and associated mechanism.

A Model Details and Baselines

A.1 Representation Model

In our experiments, all models are trained with image input in the *Particle Interactions* and *Traffic* domains. At each timestep, the observation is given as a set of image segmentations - one segmentation frame for each object in the scene. These are then transformed into object embeddings using a pre-trained CNN encoder. Each segmented frames are encoded separately, giving latent representations of each object. To allow for the capture of dynamic visual information, we concatenate two segmentation frames from consecutive time steps along the channel dimension. We pre-train the representation model using the C-SWM [24] framework, which uses a contrastive loss signal in latent space. The hyperparameters used during training of the encoder are equivalent to those reported in [24].

For the *Team Sports* domain, the ground-truth states of the players and the ball is used directly.

A.2 COMET Architecture

The architecture of COMET consists of two parts, the *mechanisms* and the *composition module*, as described in the main text. The first part, the set of mechanisms $f_{mech}^{0:M}(\cdot)$, operates on the concatenated representations of two objects, the primary object and the context object. The mechanisms are implemented as feed-forward networks. The architecture of a single mechanism is shown in Table 1. The composition module itself consists of a set of independent feed-forward confidence networks, one for each mechanism. Each such independent network receives the concatenated representations of two objects as input, similar to the mechanisms, and outputs a scalar confidence score. The architecture of a single confidence network is shown in Table 2. The source code for COMET will be released with a camera ready version of the paper.

A.3 Baselines

A.3.1 Contrastive Learning of Strutured World Models

Contrastive Learning of Strutured World Models (C-SWM) [24] is a state-of-the-art object-centric world model. At its core is a GNN transition model that takes object representations as its nodes and is assumed to be fully-connected. We use the official implementation of the model ³ but replace the default encoder with the encoder presented in Appendix A.1. The default encoder of C-SWM decomposes a single RGB-frame into a set of objects whereas our encoder operates on a set of object segmentation frames. Other than that, we use the original training procedure and model architecture as presented in [24]. This means that the only difference between the C-SWM implementation and the COMET implementation in our experiments is the use of a GNN transition model.

³<https://github.com/tkipf/c-swm>

A.3.2 Neural Production Systems

Neural Production Systems (NPS) [13] is an object-factorised world model that, like COMET, operates with a set of mechanisms instead of a monolithic transition model architecture. Furthermore, NPS also considers sparse contexts, i.e. an object’s transition is conditioned on the state of at most one other object in the observation.

The most prominent difference between NPS and our approach is that NPS jointly learns the mechanism selection, as predicted by a dot-product attention layer, and the mechanisms themselves, whereas COMET decouples the mechanism learning and the selection of mechanisms. This limits NPS’ capability of being trained on multiple environments, as different mechanisms may be active in different environments given the same state. In contrast, the competition of experts setup of COMET produces predictions for all mechanisms and updates the one with the lowest prediction error. Thus, the only requirement for COMET is that during the competition stage, enough mechanisms exist to capture all environment dynamics. The selection of mechanisms is separately trained during the environment-specific composition phase.

In our experiments, the NPS transition model replaces the GNN transition model of the C-SWM architecture, and uses the same representation model presented in App. A.1. Since no code is provided with the paper, we implemented our own version based on the detailed algorithms provided in the appendix of the original paper [13]. We verified the implementation by confirming that model performance is on par with the results presented in the original paper when evaluated on similar environments.

A.4 Hyperparameters

During training for all of the models, the batch size is set to 1024 and we use the Adam [23] optimiser with a learning rate of $1e-4$. Object embeddings are set to be 8-dimensional.

A.5 Compute Resources

All experiments were run on a cluster containing a mixture of Nvidia Quadro RTX 6000 and Nvidia Tesla V100 accelerators. Each training run involved a single GPU only. All training runs are finished within 24 hours.

B Datasets

We evaluate COMET and the baselines on three problem domains, *Particle Interactions*, *Traffic* and *Team Sport*. In each of the domains, we define a set of environments with varying dynamics. Below we provide the details of the domains.

B.1 Particle Interactions

In the *Particle Interactions* domain, each environment is defined by a set of rules. Each of these rules consists of an interaction and a condition (i.e. when the interaction should happen). The particles interact in one of 5 modes of interactions: *straight line*, *repulsion*, *attraction*, *spring* and *spiral towards centre*. Moreover, each particle is coloured. We pick 6 randomly generated environments as the training set, summarised in Table 3. The training set consists of 2000 sampled episodes from each of these environments, each with 50 frames.

For the adaptation experiments, COMET adapts to an unseen environment with 4 particles, with the *spring* interaction if the particles are of the same colour and *repel* otherwise.

B.2 Traffic

The *traffic* domain is simulated using the CARLA simulator [6]. License information can be found on <https://github.com/carla-simulator/carla>. Here, the objects are two cars, one orange and one blue, and a traffic light. In different environments, they follow different traffic rules. We train the models on three environments, 1. cars drive at normal speeds and stop at red traffic light, 2. cars drive under a much slower speed limit and stop at red traffic light, and 3. cars ignore traffic lights. In the adaptation environment, only the orange car drives at a slow speed limit. As such, the model needs to explain new behaviours such as the blue car slowing down to avoid collision with the orange car in front. The training set has 2000 episodes from each environment. The length of each episode depends on the exact scenario, which ends if both cars are out of frame or have stopped moving over 7 timesteps.

	Condition	Interaction
Environment 1	close together otherwise	repulsion straight line
Environment 2	same colour close together otherwise	spring attraction straight line
Environment 3	same colour opposite colour otherwise	repulsion spring straight line
Environment 4	same colour is blue is red otherwise	attraction spiral towards centre spiral towards centre straight line
Environment 5	same colour otherwise	repulsion spring
Environment 6	always	spiral towards centre

Table 3: Environment details for particle interactions domain.

B.3 Team Sport

The *Team Sport* domain is simulated using a modified version of STS2 [42] (<https://github.com/electronicarts/SimpleTeamSportsSimulator>). As the domain comes with only a single player type, we define additional player types according to heuristics:

B.3.1 Simple Player

If the simple player is in control of the ball it moves towards the opponent goal. If other team players are closer to the opponent goal it tries to pass the ball to them. If a simple player is in the vicinity of the opponent goal and in control of the ball, it tries to score a goal. If a different player of the own team is in control of the ball, the simple player moves towards the opponent goal. If the ball is in control of the opponent team, the simple player moves towards the midway point between the own goal and the control player to block any attempted shot. If the ball is in the air due to a pass in progress, the simple player will move towards the ball

B.3.2 Defensive Player

The defensive player inherits all offensive behaviour from the simple player. If the defensive player is not in control of the ball, it moves towards the own goal to diffuse any counter attacks.

B.3.3 Shy Player

The shy player inherits all behaviour from the defensive player but instead of moving towards the opponent goal if in control of the ball, it moves away from the opponent goal.

For all player types, we define a specific course of action as being a single mechanism. Player types may share some mechanisms (e.g. moving towards the opponent goal) but differ in others. After executing certain actions such as shooting, passing, receiving, or tackling, players are unable to perform another action within a set number of time steps. This accounts for human players also requiring a certain reaction and orientation time after executing such actions.

We defined separate mechanisms for the ball which is modeled as a separate object. If in possession, the ball traces the control player and if in the air as a result of a pass or shot, the ball follows a straight line at constant speed until it is either caught by a player or bounces off a wall to change its direction. A list of all mechanisms is depicted in Fig. 3.

We collect a total of 5,957 episodes over four different team configurations for training and evaluate our model and baselines on a configuration of two simple players in each team, a setting which was not part of the training configurations.

The state space consists of three one-hot flags indicating home team, away team, or ball. Further, we include two one-hot variables indicating which team is currently in possession of the ball (if any), a one-hot variable indicating whether the player itself is in control of the ball, a countdown timer denoting for how many time steps the player is unable to execute an action after having executed an action shortly before, as well as position and velocity information. Dimensions that apply only to players but not to the ball (e.g. the countdown timer) are present in the ball state but set to zero.

C Stability of Mechanism Disentanglement

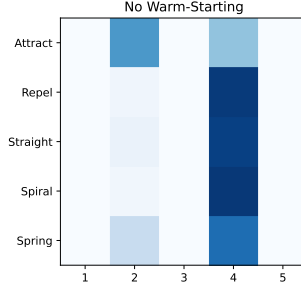


Figure 7: Disentanglement plot for training without warm-starting.

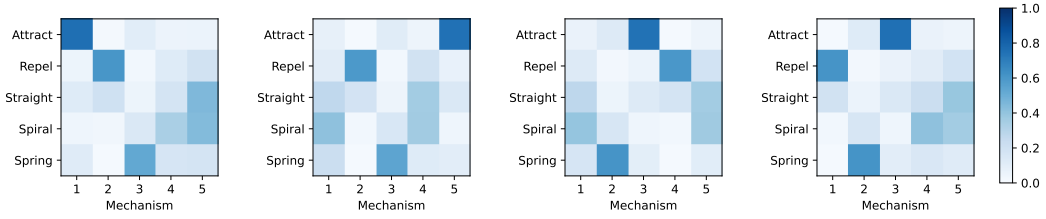


Figure 8: Disentanglement plots for different random seeds. We observe that similar structures emerge across different seeds (up to permutation).

In practice, we find that the mechanism disentanglement of COMET is significantly improved with 1) warm starting the training process and 2) increasing the time horizon for mechanism selection. With these two modifications, we find that COMET can disentangle mechanisms robustly. In Fig. 8 we show that the emergence of the discrete mechanisms is robust across different random seeds.

C.1 Warm-starting Training

One failure mode of COMET is that, due to the competitive nature of the training process, one or two mechanisms can dominate the competition and win all of the data. This can happen if the mechanisms are initialised randomly. In this case, one mechanism can, by chance, be the closest approximation to the environment dynamics and keep winning the competition. Fig. 7 shows an example of this failure case, where only two mechanisms are used. In order to alleviate this, we add a warm-start phase before the actual competitive training, where the gradient is passed through all mechanism-context pairs regardless of accuracy. The intuition is that this allows all mechanisms to be roughly correct and as such have a fair chance of specialising in particular mode of interactions.

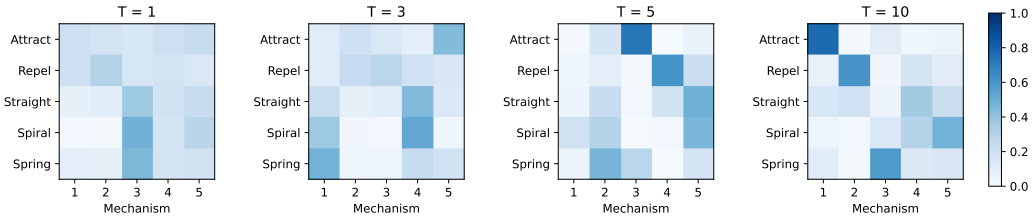


Figure 9: Effect of increasing the effective time horizon in the competition phase. As the time horizon increases, the quality of disentanglement improves.

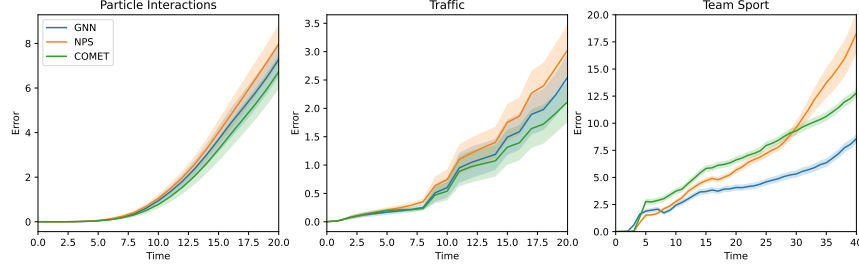


Figure 10: Given enough data, all model achieves commensurate rollout errors in the Particle Interactions and Traffic dataset. In the Team Sports dataset, COMET performs similarly to NPS, which also considers binary interactions, whereas GNN performs better. This suggests that only considering binary interactions is not sufficient in this environment. Despite this, COMET is still able to disentangle some interaction mechanisms.

C.2 Increased Time-horizon

Another failure mode that we encountered is that the model can learn non-interpretable mechanisms which leads to flickering mechanism selection during rollout. To this end, we found that increasing the effective time-horizon of the chosen mechanism-context pair is conducive to the emergence of well disentangled mechanisms. This can be implemented as a modification to the loss function, such that:

$$\mathcal{L}(\theta_{0:M}) = \sum_{i=0}^K \min_{m,j} \left[\sum_{\tau=0}^T d(\mathbf{z}_i^{t+\tau} + \Delta \mathbf{z}_i^{t+\tau}(m,j), \mathbf{z}_i^{t+\tau+1}) \right], \quad (5)$$

where T is the time horizon which is set as a hyperparameter. Note that the sum over the time horizon is inside the minimisation, meaning that a mechanism-context pair is picked only if it has the best performance over multiple consecutive time steps. This serves as an inductive bias that encourages the sparsity of mechanism changes across time steps, which empirically results in better disentangled mechanisms. Fig. 9 shows the effect of increasing the time horizon. We observe that without a long time horizon, each mechanism learns a mixture of different modes of interactions. In our experiments, we set the time horizon to 10 for Particles, 3 for Traffic and 5 for Team Sports.

D Extra Rollouts

Fig. 10 shows rollout plots for all of the models in a new environment in the large data limit. This shows that all model eventually converge to the same level of performance given enough data. Note that in the Team Sport environment is designed to be challenging for COMET as it requires reasoning about the entire scene, i.e. high order interactions between multiple objects. Here, GNN outperforms both NPS and COMET, which perform at a similar level. This supports our hypothesis that considering binary interactions can hinder model expressiveness. Nonetheless, we show that COMET is still able to disentangle mechanisms and adapt much faster than NPS, which also only considers binary interactions.