A Critical View of Vision-Based Long-Term Dynamics Prediction Under Environment Misalignment

Hanchen Xie [1 2] **Jiageng Zhu** [1 3] **Mahyar Khayatkhoei** [1] **Jiazhi Li** [1 3] **Mohamed E. Hussein** [1 4]
Wael AbdAlmageed [1 2 3]

Abstract

Dynamics prediction, which is the problem of predicting future states of scene objects based on current and prior states, is drawing increasing attention as an instance of learning physics. To solve this problem, Region Proposal Convolutional Interaction Network (RPCIN), a vision-based model, was proposed and achieved state-of-the-art performance in long-term prediction. RPCIN only takes raw images and simple object descriptions, such as the bounding box and segmentation mask of each object, as input. However, despite its success, the model's capability can be compromised under conditions of environment misalignment. In this paper, we investigate two challenging conditions for environment misalignment: *Cross-Domain* and *Cross-Context* by proposing four datasets that are designed for these challenges: *SimB-Border*, *SimB-Split*, *BlenB-Border*, and *BlenB-Split*. The datasets cover two domains and two contexts. Using RPCIN as a probe, experiments conducted on the combinations of the proposed datasets reveal potential weaknesses of the vision-based long-term dynamics prediction model. Furthermore, we propose a promising direction to mitigate the *Cross-Domain* challenge and provide concrete evidence supporting such a direction, which provides dramatic alleviation of the challenge on the proposed datasets.

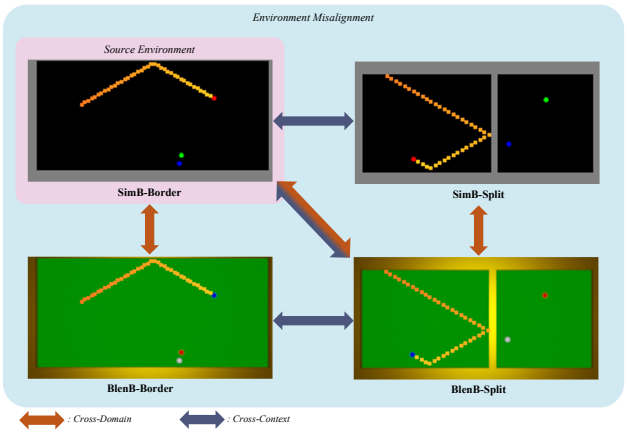

Figure 1. Illustration of two Environment Misalignment challenges: *Cross-Domain* and *Cross-Context*, and samples of four proposed datasets: *SimB-Border*, *SimB-Split*, *BlenB-Border*, and *BlenB-Split*. We adjusted images resolution and only display the ground-truth paths of the ball with displacement over time for better visualization.

1. Introduction

In addition to identifying the visual patterns observed in a scene, such as in object detection (Ren et al., 2015; Redmon et al., 2016) and segmentation (Long et al., 2015; He et al., 2017), recently, increasing attention has been drawn to learning the underlining mechanisms of scene generation, such as learning the laws of physics (Battaglia et al., 2016; Chang et al., 2017; Yi* et al., 2020; Ding et al., 2021). An intuitive instance of learning physics is *dynamics prediction* (Ye et al., 2019; Qi et al., 2021), in which the future states of discrete objects are predicted from the observations of reference states. A number of approaches has been proposed to solve the dynamics prediction problem. Object-centric models (Chang et al., 2017; Janner et al., 2019) along with interaction networks (Battaglia et al., 2016; Watters et al., 2017) focus on extracting representations and modeling dynamics of each object.

One stream of approaches models the dynamics by incorporating human defined physics models (Todorov et al.,

[1]USC Information Sciences Institute, Marina del Rey, USA [2]USC Thomas Lord Department of Computer Science, Los Angeles USA [3]USC Ming Hsieh Department of Electrical and Computer Engineering, Los Angeles, USA [4]Alexandria University, Alexandria, Egypt. Correspondence to: Hanchen Xie <hanchenx@isi.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

2012; Ullman et al., 2014; de Avila Belbute-Peres et al., 2018; Ding et al., 2021), where the physics parameters of the reference states, such as position, mass, and velocity, can be extracted from visual information or given as prior knowledge. Despite their accuracy with the guidance of Newtonian physics theory, developing a physics model that precisely describes the mechanism underlying complex real-world scenario requires expert knowledge and sophisticated physics parameters which can be hard to acquire, such as energy lost during inelastic collision. In an alternative stream of approaches, several works (Wu et al., 2015; Fragkiadaki et al., 2016; Battaglia et al., 2016; Chang et al., 2017; Wu et al., 2017; Ye et al., 2019) rely on Deep Neural Network (DNN) for learning physics, where the underline mechanism can be approximated in a data-driven way. Instead of only focusing on the abstract object state, Qi et al. (2021) propose Region Proposal Convolutional Interaction Network (RPCIN), a vision-based dynamics prediction model that takes raw image and descriptions of objects, such as bounding box and segmentation of objects which can be directly extracted from image, as input, so that it is more applicable for real-world scenario without instrumented setting (Qi et al., 2021). By adopting Region of Interest (RoI) Pooling (Girshick, 2015) and convolutional neural network (CNN), RPCIN incorporates visual information of both objects and environment, where the latter one tends to be ignored in previous works, and achieves state-of-the-art (SOTA) results, especially for long-term prediction. However, in spite of its generality to input requirements, as typical with DNNs, vision-based dynamics prediction models, like RPCIN, can be vulnerable to environment misalignment between the training and testing. In this paper, by using RPCIN as probe, we explore two types of environment misalignment challenges: *Cross-Context* and *Cross-Domain*, which can significantly compromise the model capability.

Cross-Domain environment misalignment, where the visual domain is different between training and testing, challenges the model performance, where possessing such transferability can be critical. In the real-world, although it might be easy to gather static visual information of common objects, it can be difficult to collect dynamic visual information that capture the physics properties. For example, car images are easily acquirable whereas videos of car collisions are rare. While it is possible to synthesize such scenes, transferring a model trained on synthetic data to the real world leads to performance degradation. Chang et al. (2017) postulated that the visual appearance and the dynamic properties of the object are disentangled and should be separately modeled. This postulate, therefore, narrows the discussion to be on the state space of the objects, where the inputs are semantic properties of objects rather than images, while the visual environment characteristics are ignored (Qi et al., 2021). Thus, the actual real-world challenge of shifting the visual domain of the entire environment still remains under-explored.

Cross-Context focuses on another aspect of environment misalignment challenge, where, even if the visual domain stays the same, the environment context is altered. For instance, a self-driving vehicle trained under normal traffic may suffer from irregular road condition, such as road closure with hazard sign. Most model generality discussions in previous works either ignore the changes to the environment context (Fragkiadaki et al., 2016; Ye et al., 2019; Qi et al., 2021) or represent the environment by the composition of designed objects (Battaglia et al., 2016; Chang et al., 2017; Bakhtin et al., 2019), where the relationships between objects along with different physics properties of an object, such as position, mass, and velocity, may appear during training. Furthermore, efficiently and accurately decomposing an arbitrary environment by using pre-defined objects is also challenging. Thus, we seek to directly introduce alteration to the environment context, which is distinct from varying objects composition, between training and testing stage, so that the model has to be able to correctly derive environment context characteristic by only leveraging the raw visual information.

To the best of our knowledge, there does not exist dataset benchmarks that can be used to characterize the performance of vision-based long-term dynamics prediction models, in the presence of environment misalignment challenges. To investigate the *cross-domain* and *cross-context* challenges, a set of matching datasets is necessary, where, while the visual domain or environment context is different, the underlying dynamics stay same. Thus, we propose four datasets: *SimB-Border*, *SimB-Split*, *BlenB-Border*, and *BlenB-Split*, which cover two visual domains and two environment contexts.

We show that the performance of SOTA dynamics prediction method, e.g. RPCIN, significantly degrades on either *Cross-Domain* or *Cross-Context* challenge. Further, for addressing *Cross-Domain* challenge, following Chang et al. (2017), we extend their postulate and argue that we should seek a common intermediate representation space for both object and environment. Inspired by Bakhtin et al. (2019), where the data is simply provided as semantic masks, we argue that the semantic segmentation space can serve as the intermediate space, where a visual observation model (Long et al., 2015; He et al., 2017) first maps raw images to semantic segmentation masks, and then, the masks, along with the static information of objects, are used for predicting the dynamics. Our experiments show that, on the proposed dataset with *Cross-Domain* setup, the performance by using ground-truth masks as input to RPCIN can significantly exceed the performance by using raw image as input. Even in the case that the ground-truth mask is absent, sub-optimal masks, which can be obtained via self-supervised learning, can also dramatically mitigate the *Cross-Domain* challenge.

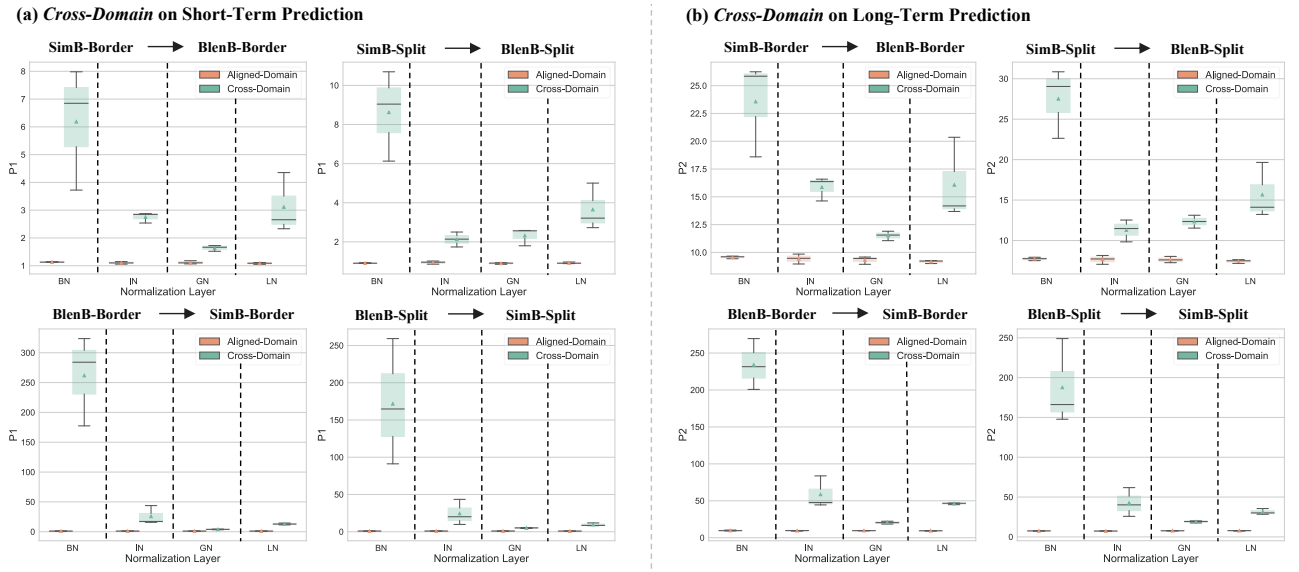

Figure 2. Performance of RPCIN with different normalization layers on *Cross-Domain* challenge. Numerical results are in Tables 2 and 3.

Our contributions can be summarised as follow: [1]

- We identify two environment misalignment challenges, *Cross-Domain* and *Cross-Context*, for vision-based long-term dynamics prediction by using RPCIN as a probe.
- We propose four datasets, *SimB-Border*, *SimB-Split*, *BlenB-Border*, and *BlenB-Split* that cover two visual domains and two environment contexts for assessing the model performance on the challenges.
- We discuss a promising direction for mitigating *Cross-Domain* challenge and provide an intuitive instance as a concretization.

2. Related Works

Dynamics Prediction is proposed as an instance of learning physics where the model is trained to derive the future states of objects or the scene given reference states as input (Battaglia et al., 2016; Chang et al., 2017; Ye et al., 2019; Janner et al., 2019; Sanchez-Gonzalez et al., 2020; Qi et al., 2021), where the dynamics model can be further applied to various tasks, such as dynamics planning (Bakhtin et al., 2019; Qi et al., 2021; Li et al., 2022) and dynamics visual reasoning (Yi* et al., 2020; Ding et al., 2021; Chen et al., 2021). Compared to considering the entire input scene as a whole, object-centric approaches (Fragkiadaki et al., 2016; Chang et al., 2017; Ye et al., 2019) focus on each object of interest individually, where the relations between objects can be derived using interaction networks (Battaglia et al., 2016; Qi et al., 2021). Instead of relying on the manually defined physics models or requiring sophisticated object physics properties as input to infer dynamics (Todorov et al., 2012; Ullman et al., 2014; Chang et al., 2017; Wu et al., 2017; de Avila Belbute-Peres et al., 2018; Ding et al., 2021), Qi et al. (2021) propose RPCIN for long-term dynamics prediction by only consuming raw images and simple object descriptions, such as bounding boxes and segmentation masks, as input, which is more feasible for real-world scenarios, and achieves SOTA performance.

Domain Adaptation has been studied on various vision tasks (Chen et al., 2018; Peng et al., 2019; Vu et al., 2019) for adapting the trained model from source visual domain to target visual domain. Literature has been created on learning to adapt with multiple supervision signal strengths (Ganin & Lempitsky, 2015; Saito et al., 2019; Shu et al., 2019), between visual domains that share various degrees of category overlapping (Zhang et al., 2018; You et al., 2019), and with source-free setting where source domain data is unavailable for adaptation (Liang et al., 2020; Liu et al., 2021). Similar with source-free setting, for better applying to nowadays real-world scenarios where the source data can be inaccessible (Liang et al., 2020), we also preclude the trained model from seeing the source data during adaptation. In conventional domain adaptation setting, where, despite vision tasks and data availability varying, the visual data itself in the target domain is sufficient for task-specific learning. Contrarily, in our proposed *Cross-Domain* challenge, model is limited from accessing the information that the physics mechanism is inherent in, such as temporal sequence. Thus, directly applying the conventional domain adaptation methods to the identified challenge may be impractical.

[1] Project is available at: https://github.com/vimal-isi-edu/VDP-EMC

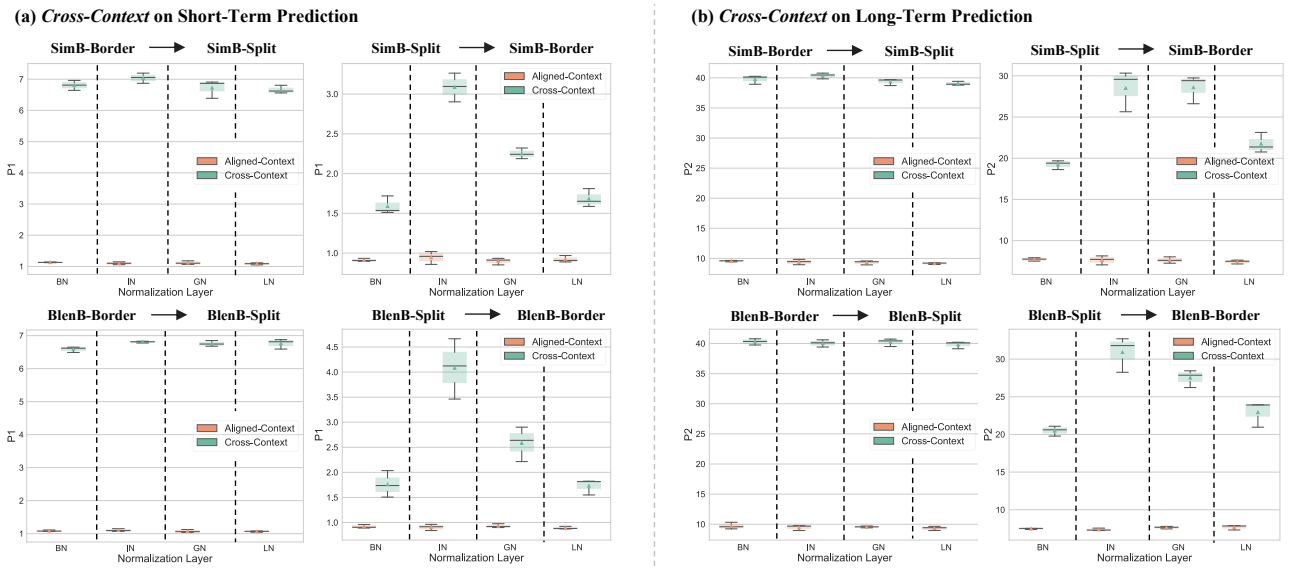

Figure 3. Performance of RPCIN with different normalization layers on *Cross-Context* challenge. Numerical results are in Appendix.

3. Method

In this section, we first introduce the problem setup of dynamics prediction and background about RPCIN, which achieves SOTA performance. Then, we propose four datasets that cover two visual domains and two environment contexts, followed by investigating *Cross-Domain* and *Cross-Context* challenges individually by using RPCIN as a case study. Finally, we discuss an encouraging direction for mitigating *Cross-Domain* challenge by mapping raw images to a common intermediate space prior to learning dynamics prediction, and discuss an intuitive instance of this scheme.

3.1. Task Formulation and Preliminary

In this paper, we focus on the dynamics prediction in billiard game scenario where the evaluations are conducted on both short-term and long-term prediction (Qi et al., 2021). During training, model takes T_{ref} consecutive image frames $\{X_{1-T_{ref}}...X_0\}$ and reference states which describe ball states in each frame $\{S_{1-T_{ref}}...S_0\}$ as input and predicts ball states of next T_{pred} frames $\{S'_1...S'_{T_{pred}}\}$, where the ground truth states $\{S_1...S_{T_{pred}}\}$ are provided as a supervision signal. During inference, the model also consumes $\{X_{1-T_{ref}}...X_0\}$ and $\{S_{1-T_{ref}}...S_0\}$ for predicting $\{S'_1...S'_{T_{pred}}\}$ and $\{S'_{T_{pred}+1}...S'_{2T_{pred}}\}$ which are evaluated as short-term and long-term predictions, respectively.

RPCIN (Qi et al., 2021) proposes to solve this problem end-to-end by extending the interaction network (Battaglia et al., 2016), and only requiring the bounding box of each ball to represent the frame state S. By utilizing RoI Pooling (Girshick, 2015), for each reference frame, each ball state feature b_i can be directly extracted from the visual feature which is encoded from the raw image through a DNN. For completeness, we will briefly summarize RPCIN here and refer the readers to Qi et al. (2021) for details. For inferring the dynamics of each ball, borrowing the notations from Qi et al. (2021), RPCIN defines five CNNs. f_O takes the state feature b_i^t of the i-th ball at the t-th timestep as input to infer its self-dynamic features. f_R pair-wisely takes b_i^t and each of the state features b_j^t of other balls at the t-th timestep as input to infer the pair-wise relative-dynamics, where the summation combines them for the total relative-dynamics. f_A takes the sum of self- and relative-dynamics features for systematically inferring the overall-dynamics features, where the results, along with the state feature b_i^t, are consumed by f_Z for a static-dynamics mixture feature. Finally, f_P takes the mixture features of the previous T_{ref} time steps to predict the state feature of the the i-th ball at the next time step. The entire process is shown in Equation (1) (Qi et al. (2021)):

$$\begin{aligned} e_i^t &= f_A(f_O(b_i^t) + \sum_{j \neq i} f_R(b_i^t, b_j^t)), \\ z_i^t &= f_Z(b_i^t, e_i^t), \\ b_i^{t+1} &= f_P(z_i^t, z_i^{t-1}, ..., z_i^{t-T_{ref}+1}) \end{aligned} \tag{1}$$

3.2. Datasets for Environment Misalignments

SimB dataset was used in RPCIN, which simulates the movements of and collisions of three balls without any information about the context of the environment, besides the image boundaries (Qi et al., 2021). Thus, we find this dataset not suitable for our usage due to its over simplicity. To this end, we propose *SimB-Border* as an extension of *SimB* by

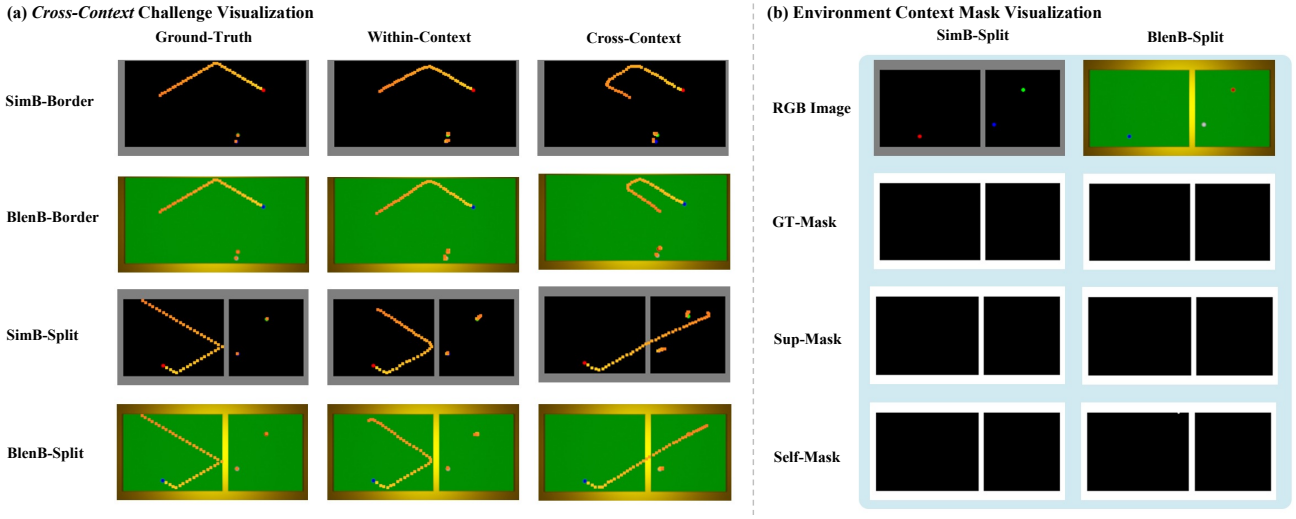

Figure 4. Visualizations of (a) *Cross-Context challenge* where trained model fails on the target context, and (b) various environment context mask. We adjusted images resolution for better visualization.

Table 1. Details of the four proposed datasets and the original SimB dataset. *Sim* stands for the visual domain of simple simulation and *Blen* stands for the visual domain rendered by Blender.

Dataset	Resolution	Domain	Context	Border Width	Split Center
SimB (Qi et al., 2021)	64×64	*Sim*	N/A	N/A	N/A
SimB-Border	192×96	*Sim*	Border	[0, 15]	N/A
SimB-Split	192×96	*Sim*	Split	[0, 15]	[64, 128]
BlenB-Border	192×96	*Blen*	Border	[0, 15]	N/A
BlenB-Border	192×96	*Blen*	Split	[0, 15]	[64, 128]

increasing the image size from 64×64 to 192×96, which includes more space for adding various contexts of environment, and introducing borders to the image boundaries. The width of the borders for the four boundaries are individually and randomly selected as integers in the range of $[0, 15]$, where zero stands for no border, and kept consistent for all frames in a single video. Introducing borders is important since it challenges the model to correctly extract the border properties from the raw image to precisely predict the locations where the ball bounces back. Additionally, to further increase the prediction difficulty and penalize the model for failing to understand the visual context, we extend *SimB-Border* to *SimB-Split* by adding a five-pixels wide vertical bar into the scene. For each video, the horizontal center of the bar is randomly selected as an integer in the range of [64, 128] and kept consistent over the video frames.

Aside from the *Sim* domain, for investigating the performance of the model under dissimilar visual domains, it is essential to create matching datasets in a different domain with closely comparable underlying dynamics. Empowered by having access to the full ground-truth environment contexts and objects information of datasets in *Sim* domain, we use Blender (Community, 2018) to create matching datasets, *BlenB-Border* and *BlenB-Split*, in *Blen* domain. In Blender, borders, split, and balls are rendered from objects where their properties, such as width, length, and locations, are adjusted by taking the ground-truth information of samples in *Sim* domain as guidance. Since Blender takes metric units of length (e.g., meter or inch) as dimension measurement, where the sizes in pixel in the rendered images are relative values controlled by multiple Blender parameters, we manually adjusted relevant factors for seeking a close match between *Sim* and *Blen* domains. More details about creating dataset in *Blen* domain are included in the Appendix. Visualizations of the four proposed datasets are shown in Figure 1 and details are include in Table 1.

3.3. Cross-Domain Misalignment

For the *Cross-Domain* challenge, we extend the problem setup described in Section 3.1 with two visual domains: Source Domain and Target Domain. During training, data on the source domain contains all information including image frames, static information of balls, and temporal sequence that connect individual frames for learning dynamics. On the target domain, the model still has access to individual frames and the static balls information, but the temporal sequence, where the physical dynamics are evident, is absent so that learning correct dynamics prediction is impractical. Furthermore, to increase the model's generality in the real-world, where the source domain data may be unavailable after domain shifts (Liang et al., 2020), we also limit the trained model from accessing the source domain data. The setup of testing data on both domains is the same as described in Section 3.1. Since the only substantial difference

Table 2. Performance of *Sim* domain trained RPCIN model on *Cross-Domain* challenge with various types of input. BN is used for all segmentation mask input training. In addition to the BN baseline, which takes RGB image as input, we also list other normalization method results as baselines for a comprehensive comparison and further demonstrating the advantage of unifying the visual domains with segmentation masks. Aligned and Cross results are the same for GT-Mask trained model because they share exactly the same data. **Bold** highlights the best results and underline highlights the second best results.

Source Dataset	SimB-Border				SimB-Split			
Target Dataset	SimB-Border (Aligned)		BlenB-Border (Cross)		SimB-Split (Aligned)		BlenB-Split (Cross)	
Eval Period	P1	P2	P1	P2	P1	P2	P1	P2
Raw RGB Image Input								
RGB-BN	1.131 ± 0.011	9.568 ± 0.121	6.185 ± 2.206	23.564 ± 4.307	0.913 ± 0.019	7.732 ± 0.208	8.622 ± 2.313	27.511 ± 4.322
RGB-IN	1.102 ± 0.045	9.426 ± 0.446	2.754 ± 0.188	15.863 ± 1.073	0.945 ± 0.082	7.641 ± 0.549	2.127 ± 0.381	11.281 ± 1.358
RGB-GN	1.117 ± 0.058	$\underline{9.323 \pm 0.346}$	1.637 ± 0.106	11.507 ± 0.425	$\mathbf{0.899 \pm 0.042}$	7.632 ± 0.386	2.315 ± 0.447	12.335 ± 0.792
RGB-LN	$\mathbf{1.085 \pm 0.033}$	$\mathbf{9.165 \pm 0.145}$	3.114 ± 1.086	16.074 ± 3.716	0.922 ± 0.042	7.433 ± 0.237	3.648 ± 1.199	15.662 ± 3.483
Segmentation Mask Input								
GT-Mask	$\underline{1.091 \pm 0.044}$	9.358 ± 0.465	$\mathbf{1.091 \pm 0.044}$	$\mathbf{9.358 \pm 0.465}$	0.916 ± 0.005	$\underline{7.431 \pm 0.511}$	$\mathbf{0.916 \pm 0.005}$	$\underline{7.431 \pm 0.511}$
Sup-Mask	1.093 ± 0.021	9.396 ± 0.285	$\underline{1.093 \pm 0.021}$	$\underline{9.397 \pm 0.286}$	0.971 ± 0.011	$\mathbf{7.372 \pm 0.089}$	0.981 ± 0.012	$\mathbf{7.422 \pm 0.095}$
Self-Mask	1.119 ± 0.037	9.604 ± 0.300	1.132 ± 0.035	9.614 ± 0.291	$\underline{0.911 \pm 0.025}$	7.837 ± 1.334	$\underline{0.959 \pm 0.020}$	8.017 ± 1.309

between datasets in *Sim* and *Blen* domains is the visual appearance, given an optimal appearance-agnostic dynamics prediction model, the performance should be comparable across visual domains. Due to the lack of sufficient information for fine tuning RPCIN model on the target domain, we directly evaluate the source domain trained model on the target domain. Furthermore, considering prior works in the domain adaptation field (Li et al., 2017; Chang et al., 2019) showing that various normalization methods in DNN may significantly affect the performance of a model when visual domain shifts occur, we conduct experiments with several normalization methods to comprehensively validate the performance of RPCIN under environment misalignment challenges. In detail, in addition to the widely used Batch Normalization (BN) (Ioffe & Szegedy, 2015), we also explore Instance Normalization (IN) (Ulyanov et al., 2016), Layer Normalization (LN) (Ba et al., 2016), and Group Normalization (GN) (Wu & He, 2018).

As shown in Figure 2, the model performance significantly decreases when the domain changes, where the impacts are more severe for applying the models trained on *Blen* domain to *Sim* domain. This might be due to the fact that, compared to extracting information from over simplified visual appearance of *Sim* domain, models are more likely to overfit to *Blen* domain so that the generality is further compromised. Furthermore, it is also noticeable that the model with BN, which are arguably the most commonly used architecture, is most vulnerable. This is due to the heavy dependence of its normalization statistics on the domain specific knowledge (Li et al., 2017). Unlike the conventional domain adaptation problem, where the visual data that contains sufficient task-specific information is available, such information specific for dynamics prediction is unavailable for the target domain in the proposed *Cross-Domain* challenge because the correct temporal sequence, where the underlying mechanism is inherent in, is missing. For instance, correctly inferring the dynamics of a ball is infeasible with merely one image of the ball that only contains the static properties of the ball rather than a sequence of images that can describe the movement of the ball.

3.4. Cross-Context Misalignment

Similar to the setup of *Cross-Domain* challenge, as described in Section 3.3, we also consider two environment contexts for *Cross-Context* challenge: Source Context and Target Context, where, during training, in source context, the model has access to all information, as described in Section 3.1. After shifting to the target context, correct temporal sequence and source context data become unavailable. The setup of testing data on both context is the same as described in Section 3.1. Unlike the model generality discussions in previous works, our proposed *Cross-Context* challenge focuses on altering the environment context, which is not represented by any object whose static information is explicitly provided for dynamics prediction. As shown in Figure 3, even when the visual domains are exactly the same, the performance of the model dramatically decreases by simply placing or removing a split bar in the middle of the environment context. As visualizations shown in Figure 4(a), for the models trained in the *Split* context, they tend to hallucinate a split bar in the middle of the image, whereas for the model trained in the *Border* context, they tend to ignore the bar. We hypothesize that such behavior might be because, in addition to predicting the state feature that only encodes object static information in the future, the model also tends to fuse auxiliary knowledge with respect to the particular environment context as a short-cut to minimize the empirical loss. For instance, in *Split* context, besides the visual

Table 3. Performance of *Blen* domain trained RPCIN model on *Cross-Domain* challenge with various types of input. BN is used for all segmentation mask input training. In addition to the BN baseline, which takes RGB image as input, we also list other normalization method results as baselines for a comprehensive comparison and demonstrating the advantage of unifying visual domains with segmentation masks. Aligned and Cross results are the same with Table 2 for GT-Mask trained model because they share exactly the same data. **Bold** highlights the best results and underline highlights the second best results.

Source Dataset	BlenB-Border				BlenB-Split			
Target Dataset	BlenB-Border (Aligned)		SimB-Border (Cross)		BlenB-Split (Aligned)		SimB-Split (Cross)	
Eval Period	P1	P2	P1	P2	P1	P2	P1	P2
Raw RGB Image Input								
RGB-BN	1.084 ± 0.023	9.713 ± 0.554	261.768 ± 75.655	233.977 ± 34.460	0.918 ± 0.037	7.478 ± 0.079	171.750 ± 84.274	187.635 ± 53.940
RGB-IN	1.103 ± 0.041	9.471 ± 0.443	25.600 ± 15.781	58.599 ± 21.827	0.906 ± 0.062	$\underline{7.368 \pm 0.168}$	24.460 ± 17.239	42.613 ± 18.159
RGB-GN	$\underline{1.075 \pm 0.045}$	9.560 ± 0.145	3.899 ± 0.526	20.425 ± 1.913	0.931 ± 0.039	7.641 ± 0.165	5.033 ± 0.575	18.970 ± 1.323
RGB-LN	$\mathbf{1.064 \pm 0.020}$	$\mathbf{9.345 \pm 0.350}$	12.969 ± 1.508	46.113 ± 1.157	$\underline{0.892 \pm 0.027}$	7.687 ± 0.321	9.518 ± 2.024	31.364 ± 3.740
Segmentation Mask Input								
GT-Mask	1.091 ± 0.044	9.358 ± 0.465	$\mathbf{1.091 \pm 0.044}$	$\underline{9.358 \pm 0.465}$	0.916 ± 0.005	7.431 ± 0.511	$\underline{0.916 \pm 0.005}$	$\underline{7.431 \pm 0.511}$
Sup-Mask	1.122 ± 0.036	$\underline{9.353 \pm 0.268}$	$\underline{1.121 \pm 0.036}$	$\mathbf{9.353 \pm 0.268}$	$\mathbf{0.891 \pm 0.027}$	$\mathbf{7.317 \pm 0.273}$	$\mathbf{0.889 \pm 0.027}$	$\mathbf{7.313 \pm 0.272}$
Self-Mask	1.136 ± 0.024	9.945 ± 0.563	1.136 ± 0.024	9.943 ± 0.560	0.914 ± 0.022	7.539 ± 0.227	0.944 ± 0.023	7.650 ± 0.224

information of the split bar, model may also be counting the prediction steps for determining the bouncing back location. Thus, in *Border* context, such overfitting can mislead the model for wrong predictions.

3.5. Unifying Visual Domains with Segmentation Masks

One of the core struggles that limits the model adaptation in the *Cross-Domain* challenge for vision-based dynamics prediction model is the absence of object dynamics information in the target domain, such as the correct temporal sequence as described in Section 3.3. However, the data for extracting the static information is available on both source and target domains. Therefore, the correct dynamics prediction is expected to be made on the target domain by feeding the aligned static information into the pretrained dynamics model, along with the proper reference temporal sequence once they become available during test. However, since the vision-based dynamics prediction model may entangle visual and dynamics information, such static information alignment can be difficult to reach without concurrently accessing the data from both domains, which is prohibited in our setup. Thus, we extend the belief held by Chang et al. (2017) to that the entire visual scene, including both objects and environment context, shall be mapped to an intermediate common abstraction space before feeding to the dynamics prediction model. In the scope of the billiard datasets discussed in this paper, since all balls being assumed to share the same physics properties, such as mass, radius, and shape, the bounding box of each ball is a sufficient object abstraction for dynamics prediction.

To seek an instance of environment context abstraction, inspired by PHYRE (Bakhtin et al., 2019), where the data is directly provided as semantic masks, we argue that the semantic segmentation map can serve as an adequate abstraction. With successes on the visual observation tasks, acquiring the segmentation maps, even in the conventional domain adaptation (Vu et al., 2019) or unsupervised setting (Ji et al., 2019), is feasible. To investigate the feasibility resolving *Cross-Domain* challenge by replacing RGB image with semantic segmentation mask as input to the dynamics prediction model, we considered the mask obtained from three sources: ground-truth mask (GT-Mask), supervised trained mask (Sup-Mask), and self-supervised trained mask (Self-Mask). For the environment context in the proposed dataset, there are two semantic classes: table, where the ball traverses, and border, including split bar, which bounds the movement of the balls. Thus, the GT-Mask is a two-channel mask, and is used as supervision signal to train the semantic segmentation model which is used to extract the Sup-Mask. For relaxing the requirement of having access to the ground-truth masks, which might be hard to gather in the real-world, we also train the segmentation model with self-supervised learning. Inspired by the method proposed by Caron et al. (2018), we use k-means to extract masks for training images which serves as the supervision signal for training the segmentation model. Visualization of various masks are shown in Figure 4(b) and the training details are included in Appendix. Experiments show that, on the proposed datasets, comparing with baseline results of various normalization methods, using GT-Mask as replacement can principally resolve the *Cross-Domain* challenge, and even using Self-Mask as replacement can achieve comparable results. Detail discussions are included in Section 4.2.

4. Experiments

In this section, we first briefly introduce the experiment details, and then provide analysis on results of *Cross-Domain* and *Cross-Context* challenges. Finally, we discuss the limitations and open issues.

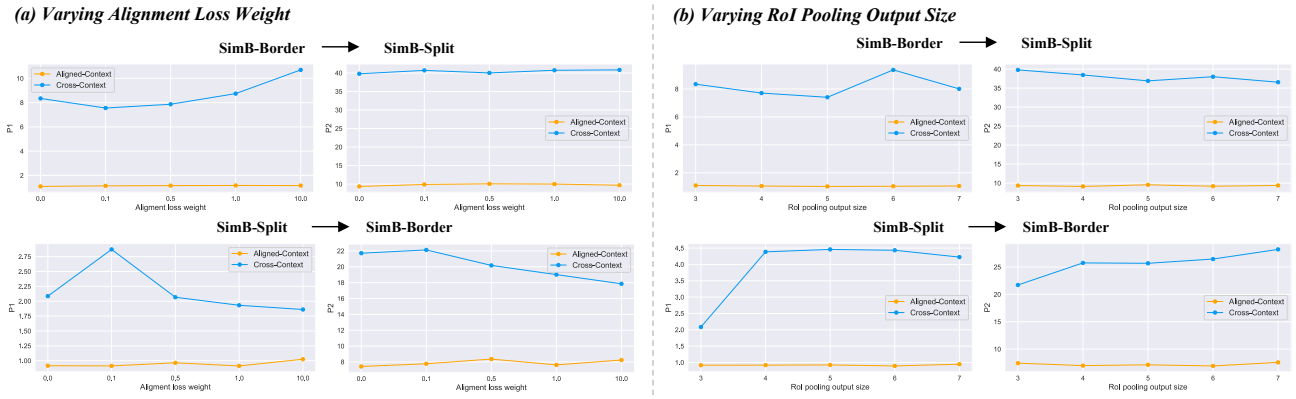

Figure 5. Ablation studies with GT-Masks as input on (a) varying alignment loss weight and (b) varying RoI pooling output size.

4.1. Training and Evaluation Details

We conduct experiments on the proposed *SimB-Border*, *SimB-Split*, *BlenB-Border*, and *BlenB-Split* datasets by using RPCIN (Qi et al., 2021) as a baseline model, and its public implementation as our code base. Following RPCIN, we use Hourglass (Newell et al., 2016) as the visual backbone, randomly flip the image horizontally and vertically as augmentation, and use the discounted loss (Watters et al., 2017) for stabilizing the training. To comprehensively evaluate the model performance, as described in Section 3.3, we replace the BN layer with GN, IN, and LN layer, where the group number in GN is set to 32. Since BN is arguably the most commonly used normalization layer, we use BN for all the training with masks as input. The RoI pooling output size is set to three. More details can be found in Appendix.

For experiment details, we set the length of reference frames T_{ref} to four and training prediction period T_{pred} to 20. Evaluations are separately conduced on short-term predictions $\{1...T_{pred}\}$ (P1) and long-term predictions $\{T_{pred}+1...2\times T_{pred)}\}$ (P2), where the squared l_2 distance between the ground-truth and predicted objects bounding box center is used as the evaluation metric. The distance is further scaled by 1000 for better demonstration. All those settings are aligned with RPCIN (Qi et al., 2021).

4.2. Discussions on *Cross-Domain*

Segmentation mask as common intermediate abstract space: In Section 3.5, we argued for mitigating *Cross-Domain* challenge by mapping the raw image to a common intermediate abstract space prior to dyanmics prediction. We also argued that the segmentation mask can serve as a promising abstract space. We conduct experiments by replacing the raw images with masks and the results are shown in Tables 2 and 3. The masks for training and testing on source and target domain are the same kind. It is noticeable that all three kinds of mask, including Self-Mask, can dramatically mitigate the *Cross-Domain* challenge and achieve a competitive performance compared with the best domain aligned results with raw images as input. Such outstanding performance empirically demonstrates the feasibility and strength of utilizing the segmentation mask as the common intermediate abstract space.

Table 4. Study of generality between various kinds of mask. Masks for training and testing on the source and target domain are obtaiend via different supervision strength. For the same environment context, GT-Masks are the same between different visual domains.

Sim Domain → Blen Domain				
	SimB-Border → BlenB-Border		SimB-Split → BlenB-Split	
	P1	P2	P1	P2
GT-Mask → GT-Mask	1.091 ± 0.044	9.358 ± 0.465	0.916 ± 0.005	7.431 ± 0.511
GT-Mask → Sup-Mask	1.091 ± 0.044	9.360 ± 0.463	0.926 ± 0.008	7.479 ± 0.494
GT-Mask → Self-Mask	1.280 ± 0.065	10.105 ± 0.381	1.236 ± 0.034	8.576 ± 0.399
Sup-Mask → Self-Mask	1.220 ± 0.035	9.776 ± 0.323	1.277 ± 0.070	8.446 ± 0.299
Blen Domain → Sim Domain				
	BlenB-Border → SimB-Border		BlenB-Split → SimB-Split	
	P1	P2	P1	P2
GT-Mask → Sup-Mask	1.091 ± 0.044	9.358 ± 0.465	0.916 ± 0.005	7.433 ± 0.511
GT-Mask → Self-Mask	1.123 ± 0.048	9.506 ± 0.447	0.962 ± 0.017	7.594 ± 0.442
Sup-Mask → Self-Mask	1.144 ± 0.038	9.468 ± 0.286	0.906 ± 0.026	7.365 ± 0.285

Generality between various kinds of mask: As shown in Figure 4(b), Sup-Mask and Self-Mask of both *Sim* and *Blen* domain are comparable with GT-Mask. Thus we further study a generality problem that the masks used for training and testing on the source and target domains are obtained via different supervision strength, which further mimic the real-world scenario where the source domain usually contains richer information than the target domain. As shown in Table 4, even though the mask obtained with weaker supervision strength may be sub-optimal, the dynamics prediction model trained on the superior mask still preserves the robustness to the *Cross-Domain* challenge.

4.3. Discussion on *Cross-Context*

Alignment loss for the predicted state feature: As described in Section 3.1, RPCIN extracts the reference ball state features b_i^t from the reference image and recursively used it to predict the ball state features b_i^{t+1} in the future. The latter one is further decoded to semantic output, such as bounding box. By replacing the raw images input with masks, future ball state features can also be extracted by utilizing the future ground-truth bounding box and reference mask, where we annotate this feature as $\hat{b}_i^{t+1}$. An intuitive path to constrain the dynamics model from encoding static irrelevant information to b^{t+1} is by introducing an alignment loss which reduces the difference between b_i^{t+1} and $\hat{b}_i^{t+1}$. We use mean-square loss as regularization with different loss weight and conduct experiments on GT-Mask. As results shown in Figure 5(a), although it seems that increasing the alignment loss weight may improve *Cross-Context* endurance on *SimB-Split* to *SimB-Border*, it does not fundamentally address the challenge and the performance gaps are still huge.

Varying RoI Pooling Output Size: The RoI pooling operation in RPCIN enables the model to consider the environment for dynamics prediction (Qi et al., 2021). By increasing the size of RoI pooling output size, richer information regarding the environment may be encoded to the ball state features that may help with enduring *Cross-Context* challenge. However, as our results on GT-Mask shown in Figure 5(b), varying RoI pooling output size does not solve the *Cross-Context* challenge.

4.4. Limitations and Open Issues

Despite the success of utilizing segmentation masks to mitigate the *Cross-Domain* challenge, the *Cross-Context* challenge is still outstanding. Even though we specifically disentangle the visual information from the dynamics information, the learned dynamics may still be entangled with other confounders, for example encoding context specific knowledge as a short-cut to minimize empirical loss. We hypothesis that, further disentanglement between underlying dynamics and latent confounders is needed to address the *Cross-Context* challenge. Further, our experiments assume all objects share the same physics properties and focus on the environment differences. However, combining alterations to both environment and object properties can further push the boundary of model generality. Finally, our experiments are conducted using synthetic data in order to obtain consistent underlying mechanisms, which already challenges the vision-based dynamics prediction model. Creating a dataset of matching synthetic and real domains is still an open issue. Although we are aware of mechanism differences, we still include experimental results with RealB (Qi et al., 2021) in Appendix for completion.

5. Conclusion

In this paper, by using RPCIN as a probe, we investigated two environment misalignment challenges: *Cross-Domain* and *Cross-Context*. Four datasets: *SimB-Border*, *SimB-Split*, *BlenB-Border*, and *BlenB-Split*, which cover two domains and two contexts, are proposed. Experiment results on the combinations of the proposed datasets reveal potential weaknesses of vison-based long-term dynamics prediction model. Furthermore, to mitigate the *Cross-Domain* challenge, we studied a promising direction and provide an intuitive instance as a concretization, whose effectiveness is demonstrated by empirical results. Lastly, we provide a discussion on the limitations and open issues.

6. Acknowledgements

This material is based on research sponsored by Air Force Research Laboratory (AFRL) under agreement number FA8750-19-1-1000. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation therein. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Laboratory, DARPA or the U.S. Government.

References

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization, 2016.

Bakhtin, A., van der Maaten, L., Johnson, J., Gustafson, L., and Girshick, R. Phyre: A new benchmark for physical reasoning. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Battaglia, P., Pascanu, R., Lai, M., Jimenez Rezende, D., and kavukcuoglu, k. Interaction networks for learning about objects, relations and physics. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Bradski, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision*, 2018.

Chang, M., Ullman, T. D., Torralba, A., and Tenenbaum, J. B. A compositional object-based approach to learning

physical dynamics. In *ICLR (Poster)*. OpenReview.net, 2017.

Chang, W.-G., You, T., Seo, S., Kwak, S., and Han, B. Domain-specific batch normalization for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Chen, Y., Li, W., Sakaridis, C., Dai, D., and Van Gool, L. Domain adaptive faster r-cnn for object detection in the wild. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

Chen, Z., Mao, J., Wu, J., Wong, K.-Y. K., Tenenbaum, J. B., and Gan, C. Grounding physical concepts of objects and events through dynamic visual reasoning. In *International Conference on Learning Representations*, 2021.

Community, B. O. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

de Avila Belbute-Peres, F., Smith, K., Allen, K., Tenenbaum, J., and Kolter, J. Z. End-to-end differentiable physics for learning and control. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Ding, M., Chen, Z., Du, T., Luo, P., Tenenbaum, J., and Gan, C. Dynamic visual reasoning by learning differentiable physics models from video and language. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 887–899. Curran Associates, Inc., 2021.

Fragkiadaki, K., Agrawal, P., Levine, S., and Malik, J. Learning visual predictive models of physics for playing billiards. In *ICLR (Poster)*, 2016.

Ganin, Y. and Lempitsky, V. S. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015.

Girshick, R. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, 2015. doi: 10.1109/ICCV.2015.169.

He, K., Gkioxari, G., Dollar, P., and Girshick, R. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 448–456. JMLR.org, 2015.

Janner, M., Levine, S., Freeman, W. T., Tenenbaum, J. B., Finn, C., and Wu, J. Reasoning about physical interactions with object-centric models. In *International Conference on Learning Representations*, 2019.

Ji, X., Henriques, J. F., and Vedaldi, A. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9865–9874, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Li, S., Wu, K., Zhang, C., and Zhu, Y. On the learning mechanisms in physical reasoning. In *NeurIPS*, 2022.

Li, Y., Wang, N., Shi, J., Liu, J., and Hou, X. Revisiting batch normalization for practical domain adaptation. In *ICLR*, 2017.

Liang, J., Hu, D., and Feng, J. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning (ICML)*, pp. 6028–6039, 2020.

Liu, Y., Zhang, W., and Wang, J. Source-free domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1215–1224, June 2021.

Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Loshchilov, I. and Hutter, F. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

Newell, A., Yang, K., and Deng, J. Stacked hourglass networks for human pose estimation. In Leibe, B., Matas, J., Sebe, N., and Welling, M. (eds.), *Computer Vision – ECCV 2016*, pp. 483–499, Cham, 2016. Springer International Publishing.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1406–1415, 2019.

Qi, H., Wang, X., Pathak, D., Ma, Y., and Malik, J. Learning long-term visual dynamics with region proposal interaction networks. In *ICLR*, 2021.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

Saito, K., Kim, D., Sclaroff, S., Darrell, T., and Saenko, K. Semi-supervised domain adaptation via minimax entropy. *ICCV*, 2019.

Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., and Battaglia, P. Learning to simulate complex physics with graph networks. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8459–8468. PMLR, 13–18 Jul 2020.

Shu, Y., Cao, Z., Long, M., and Wang, J. Transferable curriculum for weakly-supervised domain adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4951–4958, Jul. 2019. doi: 10.1609/aaai.v33i01.33014951.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

Ullman, T., Stuhlmüller, A., Goodman, N., and Tenenbaum, J. B. Learning physics from dynamical scenes. In *Proceedings of the 36th Annual Conference of the Cognitive Science society*, pp. 1640–1645, 2014.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. Instance normalization: The missing ingredient for fast stylization, 2016.

Vu, T.-H., Jain, H., Bucher, M., Cord, M., and Pérez, P. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *CVPR*, 2019.

Watters, N., Zoran, D., Weber, T., Battaglia, P., Pascanu, R., and Tacchetti, A. Visual interaction networks: Learning a physics simulator from video. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Wu, J., Yildirim, I., Lim, J. J., Freeman, B., and Tenenbaum, J. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

Wu, J., Lu, E., Kohli, P., Freeman, B., and Tenenbaum, J. Learning to see physics via visual de-animation. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Wu, Y. and He, K. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Ye, Y., Singh, M., Gupta, A., and Tulsiani, S. Compositional video prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Yi*, K., Gan*, C., Li, Y., Kohli, P., Wu, J., Torralba, A., and Tenenbaum, J. B. Clevrer: Collision events for video representation and reasoning. In *International Conference on Learning Representations*, 2020.

You, K., Long, M., Cao, Z., Wang, J., and Jordan, M. I. Universal domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Zhang, J., Ding, Z., Li, W., and Ogunbona, P. Importance weighted adversarial nets for partial domain adaptation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8156–8164, 2018. doi: 10.1109/CVPR.2018.00851.

A. Appendix

A.1. Dataset Details

In the paper, we propose four datasets: *SimB-Border*, *SimB-Split*, *BlenB-Border*, and *BlenB-Split* which cover two domains and two contexts. For the datasets on the *Sim* domain, we modified the generation code used in Qi et al. (2021) for increasing the image size from 64×64 to 192×96 and introducing borders and split bar as described in Section 3.2. 1000 videos with 100 frames in each video are generated for train and test individually. Same with Qi et al. (2021), for each video, only one ball has initial velocities whose magnitude and direction are randomly selected from $\{2, 3, 4, 5, 6\}$ and $\{\frac{i}{6}\pi | i \in [0, 1, 2, .., 11]\}$. For generating dataset in *Blen* domain, by using the ground-truth information of dataset in *Sim* domain as guidance, we represent borders, balls, and split bar as objects in Blender (Community, 2018) and adjust the object properties, such as location, width, and length, and scene properties, such as camera height, for seeking a match between datasets in *Sim* domain and *Blen* domain. We used the *Eevee* as the rendering engine in Blender.

To find a close match between *Sim* and *Blen* domains, we first create all objects, fix the camera position and direction, rendering engine, and output image resolution. Then, we iteratively adjust the scale and position of objects in Blender and check the output image until finding the unit conversion ratio between Blender (in metric) and output image (in pixel). Finally, we apply the ratio to the ground truth of each *Sim* domain sample for creating the respected sample in the *Blen* domain. However, similar to the real-world scenarios, where data is synthesised to match the real data, since the fundamental mechanisms of image sample generation are different in *Sim* and *Blen* domains, it is arguably infeasible to find an identical match. Furthermore, the slight difference between the two domains can also be considered as one aspect of the domain-specific disturbance inherent in the *Cross-Domain* challenge. By mapping various visual appearances to a common intermediate space, such disturbance can be filtered out, as shown in Tables 2 and 3.

A.2. Training Details

For the visual backbone, we followed the hourglass (Newell et al., 2016) architectures used in RPCIN (Qi et al., 2021), which is a visual feature extractor, that contains a CNN layer and three residual blocks which down-sampled the input image size by a scale of four with output channel size equals to 256, followed by a hourglass module. Details of the model architecture can be found in RPCIN (Qi et al., 2021). For normalization layer, we used BN, IN, GN, and LN, where the group number of GN is set to 32, and LN is implemented by a GN with group number set to one (Wu & He, 2018). By default, we set the output size of RoI Pooling to three and ablation study results on varying the output size are shown in Figure 5(b). Furthermore, it appears that RPCIN implemented RoI Pooling by using RoI Aligning function of PyTorch which follows (He et al., 2017).

Our experiments are conducted on two NVIDIA 1080ti GPUs. We set total batch size to 40 and models are trained over 50K iterations. We used Adam optimizer (Kingma & Ba, 2015) and set learning rate to 2×10^{-4} with cosine learning rate decay (Loshchilov & Hutter, 2017). Weight decay is set to 1×10^{-6}. For input image prepossessing, we only unified the input range to [0, 1] without using dataset specific statistics for further reducing the degree of misalignment between visual domains. Images are randomly flipped horizontally and vertically as augmentation. Same with RPCIN (Qi et al., 2021), each training sample contains four image frames and 24 frames of bounding boxes, where the first four frames of bounding boxes are used for reference and rests are used for prediction ground-truth. Similarly, each testing sample contains four image frames and 44 frames of bounding boxes for evaluating both short-term and long-term prediction. For the best use of dataset, $100 - 24 + 1 = 77$ and $100 - 44 + 1 = 57$ samples can be drawn from a single 100 frames video. Thus, for each dataset, there are 77K samples for train and 57K samples for test.

A.3. Extracting Segmentation Masks

For extracting the segmentation masks with either ground-truth label or k-means pseudo label, we adopt a simple encoder-decoder based model. We use the visual feature extractor (prior to the hourglass module) that contains a CNN layer and three residual blocks, as previously described with BN layer, as encoder, which down-samples the size of image by a scale of four with 256 channel outputs. The decoder is a simple five CNN layer with LeakyRelu as activation layer that upsamples the visual feature to the original size with two channels which represent the semantic meaning of a certain location. We use cross-entropy loss for calculating the difference between the prediction and the ground-truth. For self-supervised learning, we use k-means function of OpenCV(Bradski, 2000) to generate pseudo labels for providing supervision signal. Prior to k-means segmentation, for removing the appearance of balls from image that might be mis-classified as border, we first

blur the input image and then replace the pixels in those regions that contain balls by the respected pixels in the blurred image. The processed images are used for generating k-means pseudo label. For matching the k-means pseudo label with correct semantic meaning, we assumes that the number of pixels that represent border is less than the number of pixels that represent the table.

A.4. Additional Results

We provide numerical results on model under *Cross-Context* challenge in Tables 5 and 6, and under both *Cross-Context* and *Cross-Domain* challenges in Tables 7 and 8. For completion, we also conduct experiments on *RealB* dataset (Qi et al., 2021) and results are shown in Tables 9 and 10. However, since the underlying physics mechanism is different between *RealB* and the four proposed dataset, we find it is hard to draw any structural conclusion from the experiments and the results are only provided for reference.

Table 5. Performance of *Border* context trained RPCIN model on *Cross-Context* challenge with various types of input. Aligned and Cross results on SimB and BlenB are the same for GT-Mask trained model because they share exactly the same data.

Source Dataset	SimB-Border				BlenB-Border			
Target Dataset	SimB-Border (Aligned)		SimB-Split (Cross)		BlenB-Border (Aligned)		BlenB-Split (Cross)	
Eval Period	P1	P2	P1	P2	P1	P2	P1	P2
Raw RGB Image Input								
RGB-BN	1.131 ± 0.011	9.568 ± 0.121	6.804 ± 0.159	39.773 ± 0.726	1.084 ± 0.023	9.713 ± 0.554	6.587 ± 0.085	40.294 ± 0.518
RGB-IN	1.102 ± 0.045	9.426 ± 0.446	7.039 ± 0.164	40.364 ± 0.483	1.103 ± 0.041	9.471 ± 0.443	6.807 ± 0.028	40.054 ± 0.600
RGB-GN	1.117 ± 0.058	9.323 ± 0.346	6.721 ± 0.289	39.356 ± 0.553	1.075 ± 0.045	9.560 ± 0.145	6.759 ± 0.087	40.223 ± 0.644
RGB-LN	1.085 ± 0.033	9.165 ± 0.145	6.662 ± 0.127	39.051 ± 0.330	1.064 ± 0.020	9.345 ± 0.350	6.762 ± 0.150	39.816 ± 0.597
Segmentation Mask Input								
GT-Mask	1.091 ± 0.044	9.358 ± 0.465	8.345 ± 1.225	39.793 ± 0.917	1.091 ± 0.044	9.358 ± 0.465	8.345 ± 1.225	39.793 ± 0.917
Sup-Mask	1.093 ± 0.021	9.396 ± 0.285	8.225 ± 0.395	40.338 ± 0.445	1.122 ± 0.036	9.353 ± 0.268	9.271 ± 1.369	40.398 ± 0.805
Self-Mask	1.119 ± 0.037	9.604 ± 0.300	7.507 ± 0.237	40.639 ± 0.908	1.136 ± 0.024	9.945 ± 0.563	8.727 ± 1.476	41.788 ± 1.302

Table 6. Performance of *Split* context trained RPCIN model on *Cross-Context* challenge with various types of input. Aligned and Cross results on SimB and BlenB are the same for GT-Mask trained model because they share exactly the same data.

Source Dataset	SimB-Split				BlenB-Split			
Target Dataset	SimB-Split (Aligned)		SimB-Border (Cross)		BlenB-Split (Aligned)		BlenB-Border (Cross)	
Eval Period	P1	P2	P1	P2	P1	P2	P1	P2
Raw RGB Image Input								
RGB-BN	0.913 ± 0.019	7.732 ± 0.208	1.589 ± 0.113	19.238 ± 0.550	0.918 ± 0.037	7.478 ± 0.079	1.760 ± 0.265	20.492 ± 0.660
RGB-IN	0.945 ± 0.082	7.641 ± 0.549	3.087 ± 0.181	28.518 ± 2.517	0.906 ± 0.062	7.368 ± 0.168	4.083 ± 0.602	30.913 ± 2.351
RGB-GN	0.889 ± 0.042	7.632 ± 0.386	2.250 ± 0.067	28.597 ± 1.720	0.931 ± 0.039	7.641 ± 0.165	2.585 ± 0.346	27.511 ± 1.150
RGB-LN	0.922 ± 0.042	7.433 ± 0.237	1.683 ± 0.115	21.756 ± 1.232	0.892 ± 0.027	7.687 ± 0.321	1.730 ± 0.156	22.934 ± 1.711
Segmentation Mask Input								
GT-Mask	0.916 ± 0.005	7.431 ± 0.511	2.085 ± 0.245	21.713 ± 2.458	0.916 ± 0.005	7.431 ± 0.511	2.085 ± 0.245	21.713 ± 2.458
Sup-Mask	0.971 ± 0.011	7.372 ± 0.089	2.006 ± 0.241	20.220 ± 1.725	0.891 ± 0.027	7.317 ± 0.273	2.151 ± 0.401	21.798 ± 2.426
Self-Mask	0.911 ± 0.025	7.837 ± 1.334	2.031 ± 0.292	21.521 ± 1.375	0.914 ± 0.022	7.539 ± 0.227	2.433 ± 0.273	21.020 ± 0.127

Table 7. Performance of *Border* context trained RPCIN model on *Cross-Context* and *Cross-Domain* challenge with various types of input.

Source Dataset	SimB-Border				BlenB-Border			
Target Dataset	SimB-Border (Aligned)		BlenB-Split (Cross)		BlenB-Border (Aligned)		SimB-Split (Cross)	
Eval Period	P1	P2	P1	P2	P1	P2	P1	P2
Raw RGB Image Input								
RGB-BN	1.131 ± 0.011	9.568 ± 0.121	12.024 ± 3.017	48.670 ± 2.299	1.084 ± 0.023	9.713 ± 0.554	262.567 ± 75.581	228.936 ± 41.361
RGB-IN	1.102 ± 0.045	9.426 ± 0.446	10.359 ± 0.664	43.412 ± 0.736	1.103 ± 0.041	9.471 ± 0.443	28.746 ± 11.316	67.175 ± 13.291
RGB-GN	1.117 ± 0.058	9.323 ± 0.346	7.072 ± 0.267	40.172 ± 0.731	1.075 ± 0.045	9.560 ± 0.145	8.217 ± 0.242	42.131 ± 1.377
RGB-LN	1.085 ± 0.033	9.165 ± 0.145	7.729 ± 0.671	41.274 ± 1.092	1.064 ± 0.020	9.345 ± 0.350	14.404 ± 1.762	53.815 ± 2.412
Segmentation Mask Input								
GT-Mask	1.091 ± 0.044	9.358 ± 0.465	8.345 ± 1.225	39.793 ± 0.917	1.091 ± 0.044	9.358 ± 0.465	8.345 ± 1.225	39.793 ± 0.917
Sup-Mask	1.093 ± 0.021	9.396 ± 0.285	8.259 ± 0.394	40.414 ± 0.420	1.122 ± 0.036	9.353 ± 0.268	9.232 ± 1.356	40.303 ± 0.748
Self-Mask	1.119 ± 0.037	9.604 ± 0.300	7.712 ± 0.282	40.893 ± 0.951	1.136 ± 0.024	9.945 ± 0.563	8.252 ± 1.111	41.206 ± 1.113

Table 8. Performance of *Split* context trained RPCIN model on *Cross-Context* and *Cross-Domain* challenge with various types of input.

Source Dataset	SimB-Split				BlenB-Split			
Target Dataset	SimB-Split (Aligned)		BlenB-Border (Cross)		BlenB-Split (Cross)		SimB-Border (Cross)	
Eval Period	P1	P2	P1	P2	P1	P2	P1	P2
Raw RGB Image Input								
RGB-BN	0.913 ± 0.019	7.732 ± 0.208	9.471 ± 3.328	43.641 ± 8.771	0.918 ± 0.037	7.478 ± 0.079	171.341 ± 79.239	193.574 ± 45.798
RGB-IN	0.945 ± 0.082	7.641 ± 0.549	12.034 ± 2.199	52.677 ± 5.350	0.906 ± 0.062	7.368 ± 0.168	29.570 ± 14.694	70.004 ± 15.201
RGB-GN	0.889 ± 0.042	7.632 ± 0.386	8.250 ± 2.052	47.316 ± 3.474	0.931 ± 0.039	7.641 ± 0.165	8.721 ± 1.168	44.849 ± 3.762
RGB-LN	0.922 ± 0.042	7.433 ± 0.237	8.251 ± 3.202	46.966 ± 10.581	0.892 ± 0.027	7.687 ± 0.321	12.106 ± 1.250	57.674 ± 2.721
Segmentation Mask Input								
GT-Mask	0.916 ± 0.005	7.431 ± 0.511	2.085 ± 0.245	21.713 ± 2.458	0.916 ± 0.005	7.431 ± 0.511	2.085 ± 0.245	21.713 ± 2.458
Sup-Mask	0.971 ± 0.011	7.372 ± 0.089	2.007 ± 0.240	20.224 ± 1.723	0.891 ± 0.027	7.317 ± 0.273	2.151 ± 0.401	21.799 ± 2.426
Self-Mask	0.911 ± 0.025	7.837 ± 1.334	2.053 ± 0.291	21.519 ± 1.390	0.914 ± 0.022	7.539 ± 0.227	2.428 ± 0.274	21.021 ± 0.113

Table 9. Performance of realB trained RPCIN model on *Cross-Context* and *Cross-Domain* challenge with various types of input.

Source Dataset	RealB									
Target Dataset	RealB (Aligned)		SimB-Border (Cross)		SimB-Split (Cross)		BlenB-Border (Cross)		BlenB-Split (Cross)	
Eval Period	P1	P2	P1	P2	P1	P2	P1	P2	P1	P2
Raw RGB Image Input										
RGB-BN	0.421 ± 0.009	3.004 ± 0.091	28.604 ± 3.428	67.030 ± 4.847	37.838 ± 10.681	81.673 ± 9.417	49.760 ± 19.008	84.241 ± 19.333	52.235 ± 17.935	91.045 ± 16.429
RGB-IN	0.549 ± 0.020	3.598 ± 0.173	12.492 ± 1.948	48.657 ± 6.188	15.412 ± 1.563	56.800 ± 4.458	9.121 ± 0.370	41.226 ± 0.710	15.324 ± 0.466	55.429 ± 1.356
RGB-GN	0.423 ± 0.017	3.050 ± 0.304	9.055 ± 0.993	74.546 ± 58.718	12.413 ± 0.970	54.200 ± 3.539	8.091 ± 0.376	39.766 ± 1.036	11.413 ± 0.604	51.350 ± 1.351
RGB-LN	0.416 ± 0.038	3.163 ± 0.268	8.669 ± 2.168	40.725 ± 3.583	12.341 ± 1.688	54.137 ± 1.208	8.366 ± 1.412	41.000 ± 2.947	12.527 ± 1.596	57.103 ± 2.933

Table 10. Performance of various datasets trained RPCIN models on challenge with input as RealB.

Source Dataset	SimB-Border		SimB-Split		BlenB-Border		BlenB-Split	
Target Dataset	RealB (Cross)							
Eval Period	P1	P2	P1	P2	P1	P2	P1	P2
Raw RGB Image Input								
RGB-BN	23.697 ± 1.548	39.668 ± 6.773	14.651 ± 3.376	32.408 ± 4.780	32.291 ± 2.968	65.397 ± 16.150	28.887 ± 4.330	47.890 ± 9.987
RGB-IN	7.746 ± 0.981	29.736 ± 4.457	12.424 ± 3.594	42.159 ± 10.807	3.181 ± 0.885	16.514 ± 2.280	8.243 ± 4.815	32.189 ± 9.774
RGB-GN	2.510 ± 0.232	11.921 ± 1.321	4.791 ± 0.879	21.269 ± 2.668	2.166 ± 0.415	12.578 ± 0.332	4.683 ± 0.761	21.258 ± 2.482
RGB-LN	3.208 ± 0.947	14.067 ± 3.712	4.447 ± 0.567	23.891 ± 5.079	4.661 ± 1.928	19.314 ± 7.201	2.447 ± 0.478	14.576 ± 2.699