The Acquisition of Physical Knowledge in Generative Neural Networks

Luca Schulze Buschoff [1] **Eric Schulz** [1] **Marcel Binz** [1]

Abstract

As children grow older, they develop an intuitive understanding of the physical processes around them. Their physical understanding develops in stages, moving along developmental trajectories which have been mapped out extensively in previous empirical research. Here, we investigate how the learning trajectories of deep generative neural networks compare to children's developmental trajectories using physical understanding as a testbed. We outline an approach that allows us to examine two distinct hypotheses of human development – stochastic optimization and complexity increase. We find that while our models are able to accurately predict a number of physical processes, their learning trajectories under both hypotheses do not follow the developmental trajectories of children.

1. Introduction

More than 70 years ago, Turing (1950) famously suggested that "instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain." If we want to take Turing's proposal seriously, we have to ask ourselves: how do children learn?

Developmental psychologists have investigated children's learning in a number of different realms. One of the most well-studied is their acquisition of physical knowledge (Baillargeon, 1996; 2004; Spelke & Kinzler, 2007; Lake et al., 2017). Here, prior empirical work provides us with a precise understanding of the stages that children undergo during their cognitive development (see Figure 1A for an example). It, therefore, serves as an ideal testbed for our investigation.

In the present paper, we set out to formalize and test two distinct hypotheses of children's development. The first is the idea of *development as stochastic optimization*, which argues that cognitive development results from some form of stochastic optimization procedure (Gopnik et al., 2017; Ullman & Tenenbaum, 2020; Giron et al., 2022; Wolff, 1987). The second is the idea of *development as complexity increase*, which instead stipulates that the knowledge structures involved in human reasoning become more complex over time (Baillargeon, 2002; Binz & Endres, 2019).

First, we show how both hypotheses can be instantiated in a β-variational autoencoder (β-VAE) framework. We then probe models with different degrees of complexity and optimization on physical reasoning tasks using violation-of-expectation (VOE) methods (Piloto et al., 2018; Smith et al., 2019). Finally, we compare the learning trajectories of these artificial systems to the developmental trajectories of children.

We find that even fairly generic deep generative neural networks acquire many physical concepts. However, the order in which they acquire these concepts under both hypotheses does not align well with the acquisition order of children – neither hypothesis fully captures the learning trajectories of children. Thus, we conclude that the investigated models do not acquire their knowledge in accordance with Turing's proposal.

The remainder of this paper is organized as follows. Section 2 surveys previous literature on models of human-like physical knowledge and developmental trajectories. In Section 3, we illustrate how to instantiate the development as stochastic optimization and development as complexity increase hypotheses in the β-VAE framework. We then apply these models to different physical reasoning domains in Section 4. Section 5 concludes this report with a general discussion of our findings.

2. Related work

Building models with human-like physical knowledge has become an active research area in recent years (see Table 1

[1]MPRG Computational Principles of Intelligence, Max Planck Institute for Biological Cybernetics, Tübingen, Germany. Correspondence to: Luca Schulze Buschoff <luca.schulze-buschoff@tuebingen.mpg.de>.

Proceedings of the 39^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

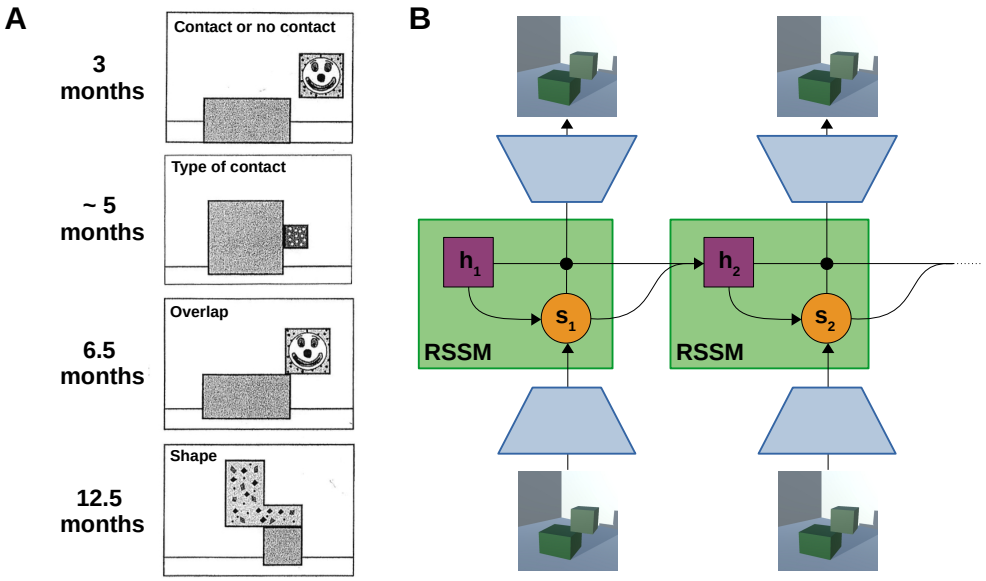

Figure 1. **A**: Human developmental trajectory for support events outlined by Baillargeon (1996). The illustrations are taken from Baillargeon (1996) and they show the physical rules acquired at the respective ages. With 3 months, infants decide based on a simple contact or no contact rule. According to this rule, a block configuration is considered stable if the blocks touch each other. At around 5 months, infants understand that the type of contact matters. Now, only configurations with blocks stacked on top of each other are judged as stable. At 6.5 months, they begin to also consider the overlap of the blocks. Finally, at 12.5 months they are able to incorporate the block shapes into their judgement, relying not only on the amount of contact but also on how the mass is distributed for each block. **B**: Illustration of our generative video prediction model.

for a summary). Battaglia et al. (2013) argued that human reasoning in complex natural scenes is driven by an intuitive physics engine that relies on probabilistic simulations to make inferences. Following this idea, they introduced interaction networks – a model that performs simulations by combining an object-centric and a relation-centric component (Battaglia et al., 2016). In contrast to the initial approach that relied on a hard-coded physics engine, interaction networks are learnable engines, allowing them to generalize to novel systems with different configurations of objects and relations. In a similar vein, Smith et al. (2019) combined a perception module that infers physical object representations from raw images with a reasoning module that predicts future object states conditioned on the object representations. They found that this model matched human performance in a number of scenarios. Lerer et al. (2016) trained large convolutional neural networks to predict the stability of wooden block towers as well as the trajectories of falling blocks. They showed that the performance of such networks exceeds that of human subjects on synthetic data. Zhang et al. (2016) compared the intuitive physics engine of Battaglia et al. (2013) to the convolutional neural network of Lerer et al. (2016). They found that while convolutional networks are able to achieve superhuman accuracy in judging the stability of block towers, their physical understanding is dissimilar to that of humans.

How physical knowledge of artificial systems should be evaluated has also received attention. Taking inspiration from developmental psychology, Piloto et al. (2018) proposed to use the VOE method to probe the knowledge of neural networks (Baillargeon, 1996). In particular, they measured the surprise of a network after observing physically implausible sequences. Their work was among the first to demonstrate that the VOE method can elucidate black-box models' inference mechanisms. Moreover, recent intuitive physics benchmarks have also been inspired by work in developmental psychology. Riochet et al. (2021) presented an "evaluation benchmark which diagnoses how much a given system understands about physics by testing whether it can tell apart well-matched videos of possible versus impossible events constructed with a game engine." Likewise, Weihs et al. (2022) proposed a benchmark testing for knowledge about continuity, solidity, and gravity using videos filmed in infant-cognition labs and robotic simulation environments. Finally, Piloto et al. (2022) also introduced a data set for evaluating intuitive physics in neural networks using the VOE method and use this data set to probe the physical knowledge of a deep learning model equipped with object-centric representations.

Even though developmental psychology has inspired how to evaluate physical knowledge in neural networks, the emphasis of prior machine learning research has always been

Table 1. Table summarizing previous work attempting to build models with human-like physical intuitions and work attempting to model human developmental trajectories. Machine learning research has predominantly focused on reproducing adult-level performance, while computational cognitive science has relied heavily on low-dimensional and static stimuli. The present paper combines the best of both worlds.

	support events	occlusion events	collision events	unsupervised learning	violation-of-expectation	sequential predictions	developmental trajectories
Work attempting to build models with human-like physical intuitions:							
Battaglia et al. (2013)	✓	✗	✗	✗	✗	✗	✗
Battaglia et al. (2016)	✗	✗	✓	✗	✗	✓	✗
Smith et al. (2019)	✗	✓	✗	✓	✓	✓	✗
Lerer et al. (2016)	✓	✗	✗	✗	✗	✗	✗
Zhang et al. (2016)	✓	✗	✗	✗	✗	✗	✗
Piloto et al. (2018)	✗	✗	✗	✓	✓	✓	✗
Riochet et al. (2021)	✗	✗	✗	✓	✓	✗	✗
Piloto et al. (2022)	✓	✓	✓	✓	✓	✓	✗
Work attempting to model human developmental trajectories:							
Giron et al. (2022)	✗	✗	✗	✗	✗	✗	✓
Averbeck (2022)	✗	✗	✗	✗	✗	✗	✓
Huber et al. (2022)	✗	✗	✗	✗	✗	✗	✓
Binz & Endres (2019)	✓	✓	✗	✗	✗	✗	✓
This work	✓	✓	✓	✓	✓	✓	✓

on reproducing adult-level performance. In contrast, computational cognitive scientists also strive to build artificial learning systems that capture the developmental trajectories of children. Perhaps most closely related to our work is the approach of Binz & Endres (2019) who compared trajectories of Bayesian neural networks that had access to different amounts of data to human developmental trajectories. They investigated both occlusion and support events and found that the acquisition order of concepts in their model aligned with that of children. However, in contrast to their work, which uses an oracle to provide a supervision signal about block stability and visibility, our approach solely relies on an unsupervised training objective.

If we look beyond the realm of intuitive physics, we can find other works that have attempted to model the process of human development. Huber et al. (2022) investigated the emergence of object recognition in children. They showed that four- to six-year-olds are already more robust to image distortions compared to deep neural networks trained on ImageNet. Furthermore, children predominantly relied on shape instead of texture for object detection, making them more similar to adults than deep neural networks (Geirhos et al., 2018). Averbeck (2022) pruned recurrent neural networks by removing weak synapses. They found that pruned networks were more resistant to distractions in a working memory task and made optimal choices more frequently in a reinforcement learning setting. These results were consistent with developmental improvements during adolescence, where performance on cognitive operations improves as excitatory synapses in the cortex are pruned. Finally, Giron et al. (2022) examined a theory of development as stochastic optimization. In particular, they combined this idea with a model of human decision-making in multi-armed bandit problems and demonstrated that development resembles a stochastic optimization process in the parameter space of this model. In contrast to these earlier models of development, our setup uses high-dimensional visual stimuli (i.e., video sequences) and solely relies on an unsupervised training objective. It, therefore, more closely resembles the actual learning processes of children in the real world.

3. Methods

In the following, we discuss how the *development as stochastic optimization* and *development as complexity increase* hypotheses can be instantiated in the β-VAE framework. For the development as stochastic optimization hypothesis, we train a generative video prediction model using gradient descent. To obtain a learning trajectory of this model, we evaluate snapshots of the model in every epoch. For the development as complexity increase hypothesis, we train models of different complexities by making use of the

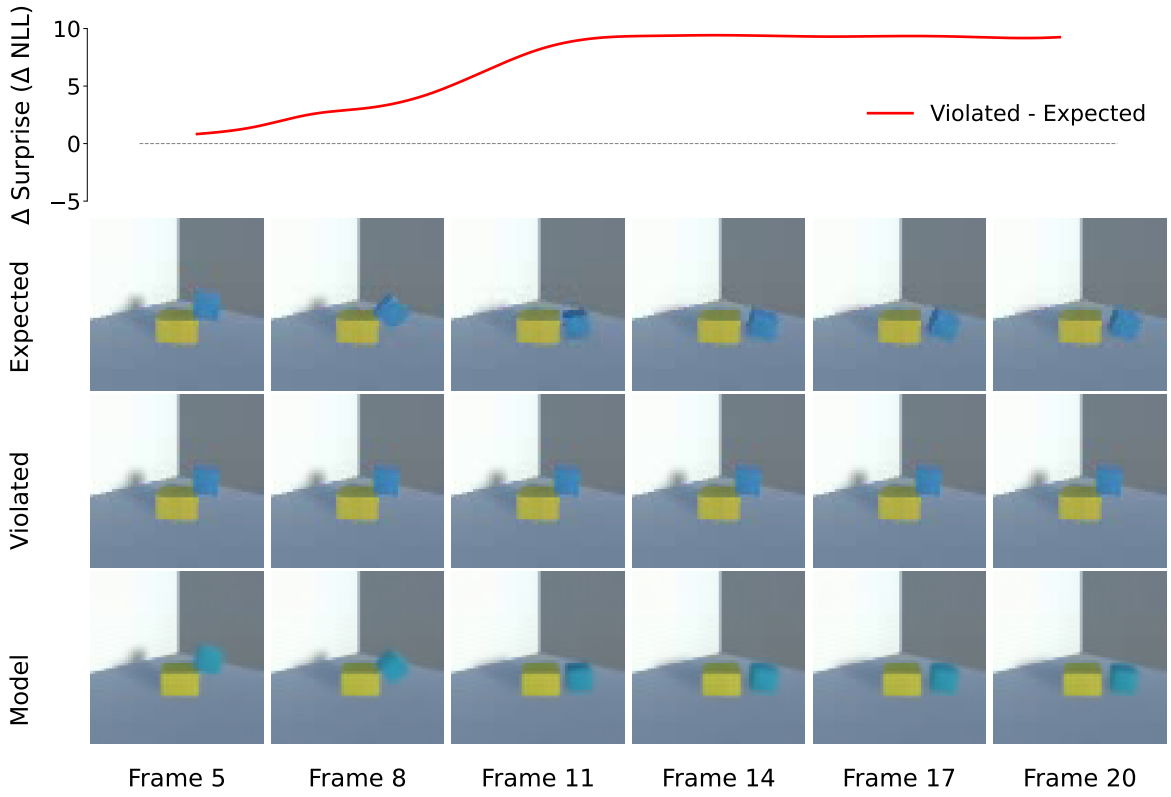

Figure 2. The first row shows the difference in surprise between the violated and expected test sequences given by the fully trained model with $\beta = 1$ for the overlap condition in the support event data-set. The surprise curve is smoothed using cubic spline interpolation. The second row shows the expected test sequence. The third row shows the violated test sequence. The last row shows the open-loop reconstruction from the model given the first two frames.

β-VAE framework (Higgins et al., 2016). Doing so enforces a bottleneck on the representational capacity of the hidden representations (Sims, 2016; Bates & Jacobs, 2020), which can be interpreted as a particular form of computational complexity (we will discuss potential alternatives in our general discussion). Learning trajectories for this hypothesis are obtained by increasing the model's representational capacity, i.e., by moving from higher to lower β-values within fully converged models.

3.1. Model architecture and objective

We use the recurrent state space model (RSSM) (Hafner et al., 2019; Saxena et al., 2021) as an exemplary model for our analysis. The RSSM can be seen as a sequential version of a VAE. It maintains a latent state at each time step, which is comprised of a deterministic component h_t and a stochastic component s_t (see Figure 1B). These components depend on the previous time steps through a function $f(h_{t-1}, s_{t-1})$, which is implemented as a gated recurrent neural network. We train our models by optimizing the following objective:

$$-\sum_{t=1}^{T} \mathbb{E}_{q(s_t|o_{\leq t})}[\ln p(o_t \mid s_t)] + \beta\, \mathbb{E}_{q(s_{t-1}|o_{\leq t-1})}\big[\mathrm{KL}\big(q(s_t \mid o_{\leq t}) \,||\, p(s_t \mid s_{t-1})\big)\big] \quad (1)$$

where $o_{\leq t} = o_1, o_2, \ldots, o_t$ is a sequence of rendered images obtained from a 3D physics engine.

For all models, the size of the stochastic hidden dimension s_t was kept at 20, while the size of the deterministic hidden dimension h_t was set to 200, as in previous implementations of the RSSM (Hafner et al., 2019; Saxena et al., 2021). We furthermore adopted the image encoder and decoder architectures described by Dittadi et al. (2020). We refer the reader to Appendix A for further details about the model architecture and training procedure.

We can use the RSSM to generate either open- or closed-loop predictions. For open-loop predictions, the model processes a number of initial observations to infer an approximate posterior $q(s_{t-1} \mid o_{\leq t-1})$, followed by decoding subsequent latent representations sampled from the prior

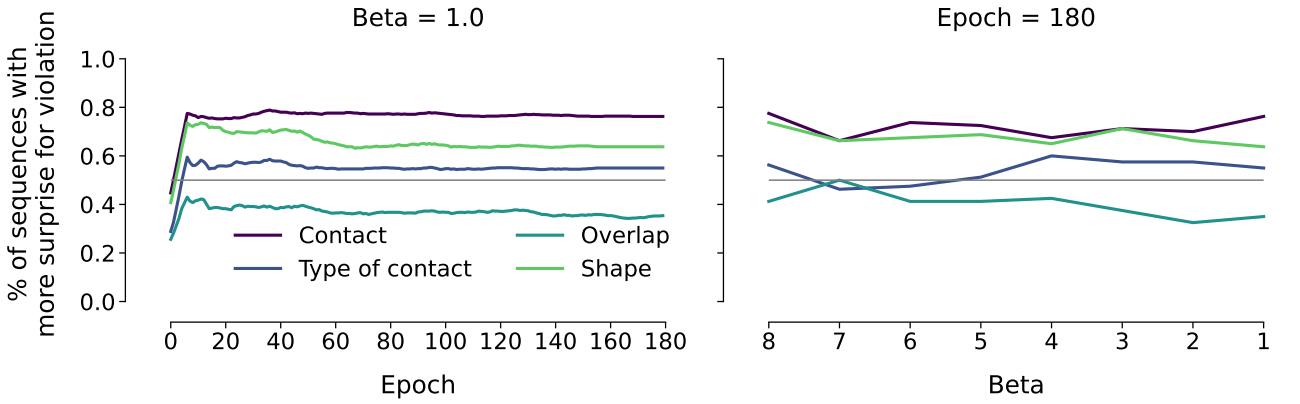

Figure 3. The plot on the left shows the percentage of sequences for which the surprise for the violated sequence exceeds that of the expected sequence for the model with $\beta = 1$ at every epoch and for each condition of the support event data set. The lines are smoothed with a uniform kernel of size 10. The plot on the right shows the same metric for fully trained models with different β.

$p(s_t \mid s_{t-1})$. For closed-loop reconstructions, the decoder is instead continuously given representations sampled from the posterior which is updated at every time step using the previously observed frame. We generally report results obtained via open-loop predictions unless stated otherwise.

3.2. Measuring surprise

To assess whether a model has learned a specific physical rule, we make use of the VOE paradigm (Piloto et al., 2018; 2022). For this, the model is presented with two video sequences: a *violated* sequence, which constitutes a violation according to the rule, and an *expected* sequence, which is consistent with the rule. If the model has successfully learned a specific rule, it should show a larger degree of surprise for the violated compared to the expected sequence. Following Piloto et al. (2022) and Smith et al. (2019), we measure the model's surprise using the negative log-likelihood (NLL) of observations under the model. More specifically, for each sequence, we determine if the NLL is larger for the violated compared to the expected sequence for the majority of the frames. We then take the mean over all sequences for a specific condition in order to check whether the reconstructions of the model better match the expected or the violated sequences. This approach is inspired by developmental psychology and allows us to measure a model's surprise similar to how developmental psychologists measure surprise in children (see Appendix B for a discussion on different measures of surprise).

4. Results

We evaluated our models on three distinct physical processes. For each of these processes, we generated training data sets inspired by experiments from developmental psychology using the Unity game engine (Unity Technologies, 2005). We randomly varied a number of properties to ensure sufficient variability in the training data. We also generated test data sets that – following the VOE paradigm – contain pairs of violated and an expected sequences for each of the conditions in the respective event types (see Appendix C for a detailed description of the data generation process and a visualization of the employed test sequences).

4.1. Support events

We began our investigations by looking at support events, which consist of block configurations such as the ones shown in Figure 1A. Similar tasks have been studied extensively in both the machine learning and developmental psychology community, and they, therefore, serve as an ideal starting point for our analyses. Each scene in our data set contains two randomly configured blocks in a gray room.[1]

Baillargeon (1996) has shown that, as infants grow older, they make use of increasingly complex rules to decide whether a given block configuration is stable or not (also see Baillargeon (2002; 2004)). With 3 months, infants decide based on a simple contact or no contact rule. According to this rule, a block configuration is considered stable if the blocks touch each other. At around 5 months, infants understand that the type of contact matters. Now, only configurations with blocks stacked on top of each other are judged as stable. At 6.5 months, they begin to also consider the overlap of the blocks. Finally, at 12.5 months they are able to incorporate the block shapes into their judgement, relying not only on the amount of contact but also on how the mass is distributed for each block.

[1]Note that it would certainly be possible to consider more complex configurations (e.g., by increasing the number of blocks), but we deliberately made this design choice to match the experimental paradigms used in developmental psychology.

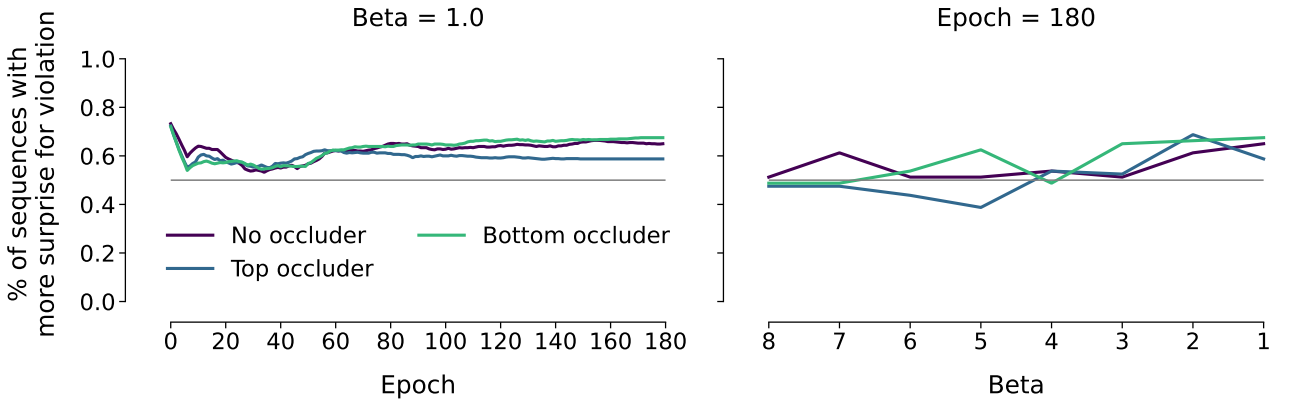

Figure 4. The plot on the left shows the percentage of sequences for which the surprise for the violated sequence exceeds that of the expected sequence for the model with $\beta = 1$ at every epoch and for each condition of the occlusion event data set. The lines are smoothed with a uniform kernel of size 10. The plot on the right shows the same metric for fully trained models with different β.

For each of the four rules for support events, we constructed pairs of violated and expected test sequences with identical first frames. For example, according to the overlap rule, a block configuration should only be stable if the blocks are stacked on top of each other with enough overlap. A test sequence pair for this rule shows two blocks that only slightly overlap (see Figure 2). The expected test sequence, which is consistent with the rule, shows the top block falling. In contrast – and in violation of real physics – the violated test sequence shows a block configuration that appears stable.

We first verified that our model is able to predict a given scene accurately into the future. For this purpose, we plotted the open-loop predictions given by the fully trained model with $\beta = 1$. Figure 2 shows an examplary result for the overlap condition. We see that the predictions of the fully trained model closely match the expected sequence. Furthermore, we see that high surprise values for the violated sequence coincide with differences to the expected sequence – the model is surprised when it observes parts of a video sequence that diverge from real physics. Appendix D.1 shows further examples for open- and closed-loop predictions, while Appendix E contains a visualization of prediction errors.

Figure 3 illustrates how knowledge about physical rules develops over time for the two earlier outlined hypotheses. On the left, the percentage of sequences for which the surprise for the violated sequence exceeds that of the expected sequence is plotted for each of the four conditions over the course of training for the model with $\beta = 1$. Here, the model becomes increasingly optimized over the epochs, thereby implementing the development as stochastic optimization hypothesis. It is evident that the model is able to learn three of the four conditions as it shows more surprise for the violated than the expected sequences for the majority of the cases. However, it learns the conditions at roughly the same rate which does not match the developmental trajectories of children. While it settles at different levels for the conditions, the order of these conditions also does not match the acquisition order of children: the shape condition, for instance, shows the second highest percentage while it is the last rule that children acquire.

On the right, the percentage of sequences for which the surprise for the violated sequence exceeds that of the matching expected sequence is plotted for each of the four conditions for fully trained models with different β-values. This relates to the development as complexity increase hypothesis since the representational capacity of the model increases as β decreases. The order in which increasingly complex models learn the different conditions again does not resemble the developmental trajectories of children: the model with $\beta = 8$ performs very similarly to the model with $\beta = 1$. To summarize, for support events, neither hypotheses yields learning trajectories that resemble the developmental trajectories of children (see also Appendix F for a closed-loop counterpart to Figure 3).

4.2. Occlusion events

Next, we wanted to test whether the results obtained in the last section hold across domains. Thus, we extended our analyses to occlusion events, which display a moving object passing behind two vertical columns. The two columns, together with an optional horizontal connection at the top or bottom, form an occluder which may hide the moving object. Like in the preceding section, we created a randomized training data set alongside several test sequences that violate physical principles (see Appendix C for examples and further details about the data generation process).

Baillargeon (1996) reported that, in this setting, (1) infants form a simple behind/not-behind distinction by 2.5 months. Hereby they assume that the object will not re-appear in the

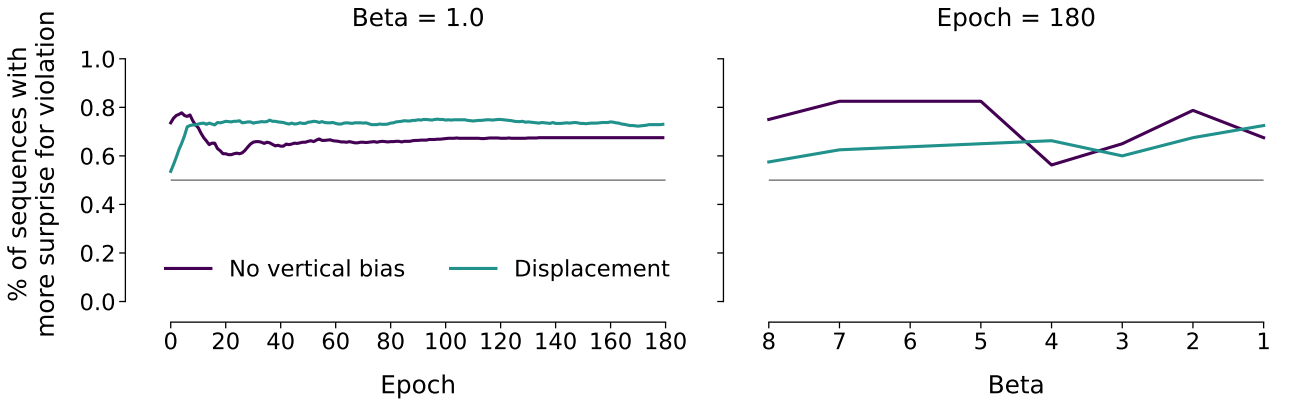

Figure 5. The plot on the left shows the percentage of sequences for which the surprise for the violated sequence exceeds that of the expected sequence for the model with $\beta = 1$ at every epoch and for each condition of the collision event data set. The lines are smoothed with a uniform kernel of size 10. The plot on the right shows the same metric for fully trained models with different β.

gap between the columns that are connected at the top or bottom. By 3 months, (2) infants expect objects to re-appear when the columns are connected at the top but fail to do so if the columns are connected at the bottom. Finally, at 3.5 months, (3) they also expect objects to appear behind screens that are connected at the bottom, given that the object is taller than the connecting part.

Figure 4 visualizes our modeling results. First, we can observe that the fully trained model with $\beta = 1$ is surprised when presented with any of the test sequences that violate physical principles, indicating that it understood all of the three aforementioned occlusion settings. For the left side of the plot, which depicts the development as stochastic optimization hypothesis, we see that the number of sequences for which the surprise for the violated sequence exceeds that of the matching expected sequence increases at the same rate for all three conditions, meaning that the model learns the three concepts at approximately the same time. Thus, for occlusion events, the stochastic optimization hypothesis again does not yield a learning trajectory that matches that of children.

The right side of Figure 4 relates to the development as complexity increase hypothesis. Here, we see that the the percentage of sequences for which the surprise for the violated sequence exceeds that of the matching expected sequence remains relatively stable as the complexity of the model increases. This again does does not lend support to the complexity increase hypothesis.

4.3. Collision events

The last physical process that we investigated were collision events. Here, each scene shows an object rolling down a hill and colliding with a stationary object. To train our models, we again created a randomized training data set of such scenarios together with several test sequences that violate physical principles (see Appendix C for examples and further details about the data generation process).

Baillargeon (1996) provides two insights when it comes to collision events: (1) at first, infants expect any stationary object that collides with a moving object to be displaced by the same amount. However, as they grow older, they are able to take the relative sizes of the two objects into account and understand that the larger the size of the moving object compared to the stationary object, the larger the displacement of the stationary object. (2) Furthermore, at around 8 months, infants become subject to a *vertical bias*, meaning that they judge stationary objects as immovable if they have a salient vertical dimension (Wang et al., 2003; 2004).

Figure 5 again shows the learning trajectories of the two hypotheses. Empirical research suggests that children first expect a size-independent displacement for all objects. To test whether our models exhibit such a characteristic, we compared their predictions for violated sequences with size-independent displacement (thereby violating physical principles) and expected sequences with a size-dependent displacement (working according to normal physics). While children initially show higher surprise when observing the expected sequences, our models offer a very different picture: at no point do they show more surprise for the expected compared to the violated sequences, as apparent by the plot on the left side of Figure 5.

To test for the vertical bias, we constructed violated sequences where vertical objects do not move upon a collision and expected sequences where they do move according to normal physics. When presented with such sequence pairs, children show more surprise for the expected compared to the violated sequences at some point during their development. However, our models do not exhibit this characteristic

at any point in time. Throughout training, they are more surprised by the violated compared to the expected sequences. Likewise, we also do not observe this effect when manipulating the representational capacity of our models as shown on the right side of Figure 5. For collision events, we therefore again find that neither the development as stochastic optimization nor the development as complexity increase hypothesis yield learning trajectories that resemble the developmental trajectories of children.

5. Discussion

We have compared the learning trajectories of an artificial system to the developmental trajectories of children for three physical processes. For this purpose, we outlined an approach that allowed us to investigate two distinct hypotheses of human development: stochastic optimization and complexity increase.

We found that the learning trajectories under both hypotheses do not follow children's developmental trajectories. For all three event types, we found differences to human learning. For support and occlusion events, the predictions of our models improve at roughly the same rate for all conditions, which indicates that our models do not move along separate stages. For collision events, our models crucially exhibit none of the biases that appear in children. We argue that this is to be expected. The vertical bias, for example, is likely a product of their self-directed movement in the world: as children begin to move around, the majority of vertical objects they encounter, such as walls or furniture, are immovable (Baillargeon, 1996). In contrast to this, our models do not have access to such experiences and are therefore not incentivized to show this bias.

While previous work on modeling cognitive development (Binz & Endres, 2019; Giron et al., 2022) focused on tasks with low-dimensional and static stimuli, our approach employs high-dimensional visual stimuli (e.g., video sequences) and solely relies on an unsupervised training objective. It, therefore, more closely mirrors the actual learning processes of children in the real world. We furthermore extend previous research on building models with human-like physical intuitions by not focusing on adult-level performance but instead investigating developmental trajectories (see again Table 1 for a comparison to previous research).

To showcase how our approach functions as a general framework for testing the learning trajectories of artificial systems, we used a fairly generic generative model. It would be interesting to evaluate the two hypotheses for other model classes, such as generative adversarial networks (Goodfellow et al., 2020) or diffusion models (Sohl-Dickstein et al., 2015). Furthermore, it has been argued in previous work that object-centric representations are crucial for a proper physical understanding of more complex scenes (Piloto et al., 2022). However, our models did not feature explicit object-centric representations and were still able to predict a number of physical processes. Thus, future work should aim for a systematic comparison of models with and without explicit object-centric representations.

We used very simple data sets to determine the viability of our approach. Evidently, children do not learn by looking at a large number of stylized sequences. Instead, they observe the real world and generalize their acquired knowledge to a given experimental setting. To capture this process, future research should ideally train models in a similar way. This could, for example, be accomplished by utilizing the SAYCam data set, which contains a large number of longitudinal video recordings from infants' perspectives (Sullivan et al., 2021). We believe that using this data set, it might be possible for an artificial model to acquire a vertical bias. It additionally includes time stamps indicating when a child has encountered a particular scene, which could be used to investigate how the nature of the training data influences development.

Finally, the complexity constraint we impose is a constraint on the size of the latent representations of the model. However, it is entirely possible that other parts of children's physical models change in complexity throughout their development. For example, Binz & Endres (2019) implement complexity increase through varying the complexity of model weights instead of the complexity of latent representations. In contrast to our work, they found that the acquisition order of concepts in their model aligned with that of children for support and occlusion events. Future research should therefore also investigate different complexity constraints and the resulting learning trajectories.

What do we make of our results on the whole? On the one hand, they demonstrate that it is possible to use tools developed in psychology to elucidate the inner workings of deep learning models (Ritter et al., 2017; Binz & Schulz, 2022). From this perspective, our work highlights yet another mismatch between human learning and learning in artificial neural networks (Flesch et al., 2018; Dekker et al., 2022). On the other hand, our results also indicate that current modeling approaches are quite far away from implementing Turing's proposal for obtaining a programme that simulates the adult mind. If we want to keep following this direction, we have to therefore ask ourselves what is needed to build models that acquire their knowledge in human-like ways. Towards this end, it is possible that the training data plays an important role, as suggested by some of our results. However, it might be equally plausible that we need to develop new model architectures and come up with more sophisticated ways to train them.

References

Averbeck, B. B. Pruning recurrent neural networks replicates adolescent changes in working memory and reinforcement learning. *Proceedings of the National Academy of Sciences*, 119(22):e2121331119, 2022.

Baillargeon, R. Infants' understanding of the physical world. *Journal of the Neurological Sciences*, 143(1-2):199–199, 1996.

Baillargeon, R. The acquisition of physical knowledge in infancy: A summary in eight lessons. *Blackwell handbook of childhood cognitive development*, 1(46-83):1, 2002.

Baillargeon, R. Infants' physical world. *Current Directions in Psychological Science*, 13(3):89–94, 2004.

Baldi, P. and Itti, L. Of bits and wows: A bayesian theory of surprise with applications to attention. *Neural Networks*, 23(5):649–666, 2010.

Bates, C. J. and Jacobs, R. A. Efficient data compression in perception and perceptual memory. *Psychological review*, 127(5):891, 2020.

Battaglia, P., Pascanu, R., Lai, M., Jimenez Rezende, D., et al. Interaction networks for learning about objects, relations and physics. *Advances in Neural Information Processing Systems*, 29, 2016.

Battaglia, P. W., Hamrick, J. B., and Tenenbaum, J. B. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110 (45):18327–18332, 2013.

Binz, M. and Endres, D. Emulating human developmental stages with bayesian neural networks. *arXiv preprint arXiv:1902.07579*, 2019.

Binz, M. and Schulz, E. Using cognitive psychology to understand gpt-3. *arXiv preprint arXiv:2206.14576*, 2022.

Dekker, R. B., Otto, F., and Summerfield, C. Curriculum learning for human compositional generalization. *Proceedings of the National Academy of Sciences*, 119(41): e2205582119, 2022.

Dittadi, A., Träuble, F., Locatello, F., Wüthrich, M., Agrawal, V., Winther, O., Bauer, S., and Schölkopf, B. On the transfer of disentangled representations in realistic settings. *arXiv preprint arXiv:2010.14407*, 2020.

Flesch, T., Balaguer, J., Dekker, R., Nili, H., and Summerfield, C. Comparing continual task learning in minds and machines. *Proceedings of the National Academy of Sciences*, 115(44):E10313–E10322, 2018.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

Giron, A. P., Ciranka, S., Schulz, E., van den Bos, W., Ruggeri, A., Meder, B., and Wu, C. M. Developmental changes in learning resemble stochastic optimization. *PsyArXiv*, 2022.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Gopnik, A., O'Grady, S., Lucas, C. G., Griffiths, T. L., Wente, A., Bridgers, S., Aboody, R., Fung, H., and Dahl, R. E. Changes in cognitive flexibility and hypothesis search across human life history from childhood to adolescence to adulthood. *Proceedings of the National Academy of Sciences*, 114(30):7892–7899, 2017.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pp. 2555–2565. PMLR, 2019.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. *Open Review*, 2016.

Huber, L. S., Geirhos, R., and Wichmann, F. A. The developmental trajectory of object recognition robustness: children are like small adults but unlike big deep neural networks. *arXiv preprint arXiv:2205.10144*, 2022.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017.

Lerer, A., Gross, S., and Fergus, R. Learning physical intuition of block towers by example. In *International Conference on Machine Learning*, pp. 430–438. PMLR, 2016.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

Piloto, L., Weinstein, A., TB, D., Ahuja, A., Mirza, M., Wayne, G., Amos, D., Hung, C.-c., and Botvinick, M. Probing physics knowledge using tools from developmental psychology. *arXiv preprint arXiv:1804.01128*, 2018.

Piloto, L., Weinstein, A., Battaglia, P., and Botvinick, M. Intuitive physics learning in a deep-learning model inspired by developmental psychology. *Nature Human Behavior*, 2022.

Riochet, R., Castro, M. Y., Bernard, M., Lerer, A., Fergus, R., Izard, V., and Dupoux, E. Intphys 2019: A benchmark for visual intuitive physics understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44 (9):5016–5025, 2021.

Ritter, S., Barrett, D. G., Santoro, A., and Botvinick, M. M. Cognitive psychology for deep neural networks: A shape bias case study. In *International conference on machine learning*, pp. 2940–2949. PMLR, 2017.

Saxena, V., Ba, J., and Hafner, D. Clockwork variational autoencoders. *arXiv preprint arXiv:2102.09532*, 2021.

Sims, C. R. Rate–distortion theory and human perception. *Cognition*, 152:181–198, 2016.

Smith, K., Mei, L., Yao, S., Wu, J., Spelke, E., Tenenbaum, J., and Ullman, T. Modeling expectation violation in intuitive physics with coarse probabilistic object representations. *Advances in Neural Information Processing Systems*, 32, 2019.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Spelke, E. S. and Kinzler, K. D. Core knowledge. *Developmental Science*, 10(1):89–96, 2007.

Sullivan, J., Mei, M., Perfors, A., Wojcik, E., and Frank, M. C. Saycam: A large, longitudinal audiovisual dataset recorded from the infant's perspective. *Open Mind*, 5: 20–29, 2021.

Turing, A. M. Computing machinery and intelligence. *Mind*, LIX(236):433–460, 1950.

Ullman, T. D. and Tenenbaum, J. B. Bayesian models of conceptual development: Learning as building models of the world. *Annual Review of Developmental Psychology*, 2020.

Unity Technologies. Unity game engine. 2005.

Wang, S.-h., Kaufman, L., and Baillargeon, R. Should all stationary objects move when hit? developments in infants' causal and statistical expectations about collision events. *Infant Behavior and Development*, 26(4):529–567, 2003.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Weihs, L., Yuile, A. R., Baillargeon, R., Fisher, C. L., Marcus, G., Mottaghi, R., and Kembhavi, A. Benchmarking progress to infant-level physical reasoning in ai. *Manuscript under review*, 2022.

Wolff, J. Cognitive development as optimisation. In *Computational Models of Learning*, pp. 161–205. Springer, 1987.

Zhang, R., Wu, J., Zhang, C., Freeman, W. T., and Tenenbaum, J. B. A comparative evaluation of approximate probabilistic simulation and deep neural networks as accounts of human physical scene understanding. *arXiv preprint arXiv:1605.01138*, 2016.

A. Model implementation and training details

The models were implemented in PyTorch (Paszke et al., 2019). For all models, the size of the stochastic hidden dimension s_t was kept at 20, while the size of the deterministic hidden dimension h_t was set to 200, as in previous implementations of the RSSM (Hafner et al., 2019; Saxena et al., 2021).

We used the encoder and decoder from Dittadi et al. (2020). The encoder consists of 3 blocks. The first block consists of a convolutional layer with a kernel of size 5 and a stride of 2 and a padding of 2, followed by a leaky ReLU activation function, followed by 2 residual blocks. The second block consists of a convolutional layer with a kernel of size 1 and a stride of 1 and no padding, followed by average pooling with a kernel of size 2, followed by 2 blocks residual blocks. The third block consists of average pooling with a kernel of size 2, followed by 2 residual blocks. The fourth block consists of a convolutional layer with a kernel of size 1 and a stride of 1 and no padding, followed by average pooling with a kernel of size 2, followed by 2 residual blocks. The fifth block consists of average pooling with a kernel of size 2, followed by 2 residual blocks.

The decoder consists of 5 blocks. The first block consists of 2 residual blocks, followed by upsampling with a scale factor of 2. The second block consists of 2 residual blocks, followed by a deconvolutional layer with a kernel size of 1 and a stride of 1, followed by upsampling with a scale factor of 2. The third block again consists of 2 residual blocks, followed by upsampling with a scale factor of 2. The fourth block consists of 2 residual blocks, followed by a deconvolutional layer with a kernel size of 1 and a stride of 1, followed by upsampling with a scale factor of 2. The fifth block consists of 2 residual blocks, followed by upsampling with a scale factor of 2, a leaky ReLU activation funktion, followed by a deconvolutional layer with a kernel size of 5 and a stride of 1 and a padding of 2.

The models were trained for 180 epochs using a batch size of 32. The loss function was optimized using the Adam optimiser with a learning rate of 0.001 (Kingma & Ba, 2014), which was divided by 10 every 50 epochs. The models were trained on a NVIDIA Quadro RTX 5000 for roughly 7 days. Our implementation of the RSSM borrows from a previous implementation on GitHub. The complete code for this project, including our model implementation, is available upon request.

B. Different measures of surprise

Smith et al. (2019) measure surprise as the maximum of the negative log-likelihood of observations under the model. Likewise, Piloto et al. (2022) use the sum of the squared error, which given a Gaussian distribution with a standard deviation of one also equals the negative log-likelihood of observations under the model, up to a constant. We report the same measure in the main paper. However, Piloto et al. (2018) propose another surprise measure: the KL-divergence between the prior and posterior over the latent representation (Baldi & Itti, 2010). We confirmed our results using this measure and found only slight differences between the two measures (see Figures 6 and 7 as an example).

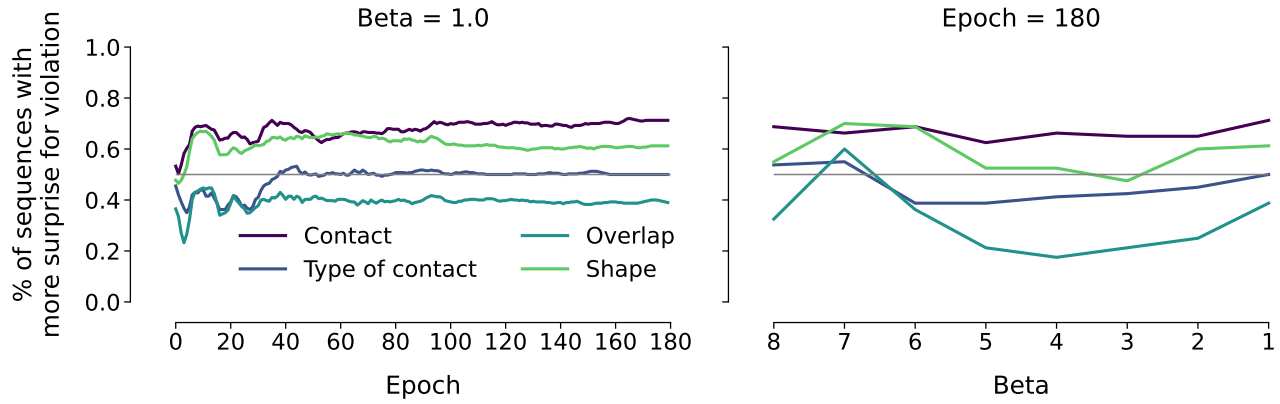

Figure 6. Replication of Figure 3 using the KL-based surprise measure. The plot on the left shows the percentage of sequences for which the surprise for the violated sequence exceeds that of the expected sequence for the model with $\beta = 1$ at every epoch and for each condition of the support event data set separately. The lines are smoothed with a uniform kernel of size 10. The plot on the right shows the same metric for fully trained models with different β.

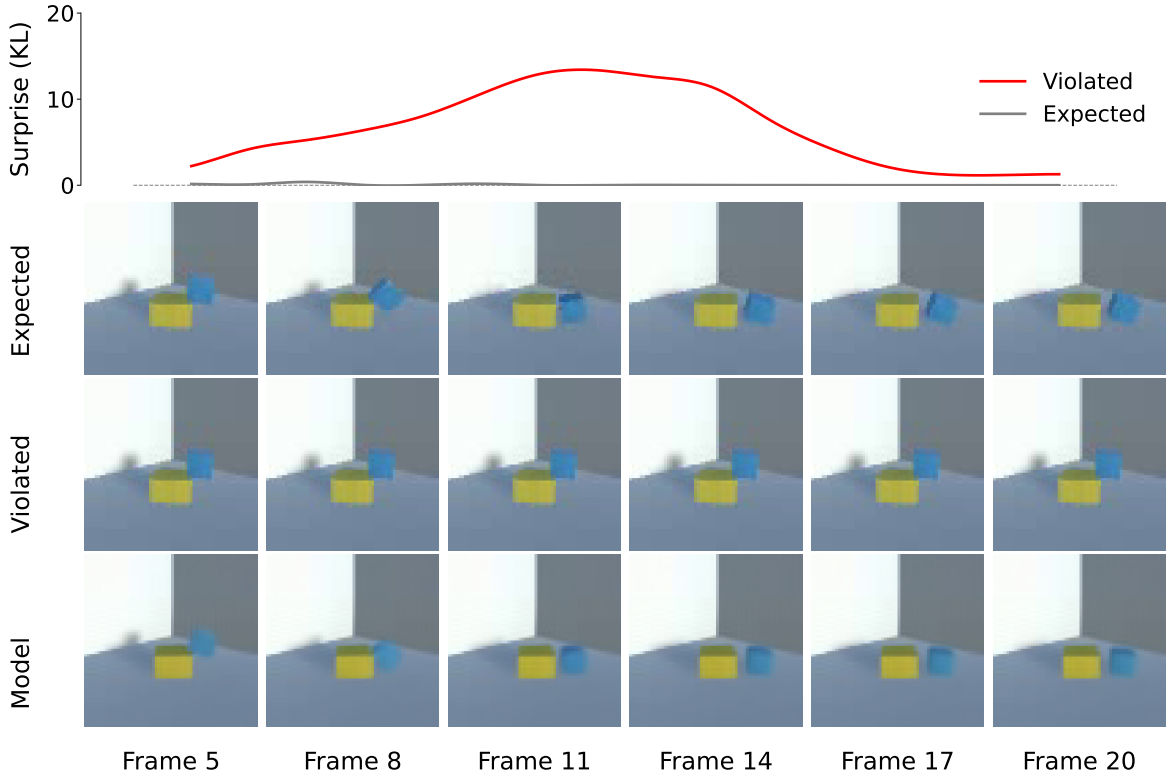

Figure 7. Replication of Figure 2 using the KL-based surprise measure. The first row shows the surprise given by the fully trained model with $\beta = 1$ for the displayed expected and violated sequence. The surprise curves are smoothed using cubic spline interpolation. The second row shows the expected test sequence. The third row shows the violated test sequence. The last row shows the open-loop reconstruction from the model. The first four frames were removed for this plot.

C. Training and test data sets

Each of the three event types is split into a training data set and a test data set. The training data set features 100.000 video sequences which each consist of 20 frames with a size of [64, 64, 3]. It was randomly split into 99.000 training sequences and 1000 validation sequences. The test data sets feature 80 pairs of expected and violated video sequences for each of the individual conditions in the respective event types. For the support event types, this results in a test data set with 640 video sequences. For the occlusion event types, the test data set consists of 480 video sequences. Finally, the test data set for the collision event types features 320 video sequences. The video sequences again consist of 20 frames with a size of [64, 64, 3].

For the support events, the following variations were performed in order to ensure sufficient variability in the data sets: lower block size, lower block color, upper block color, lower block rotation, upper block rotation, upper block position (offset), and camera angle (see Figure 8 for more exemplary test sequences). For the training data set the shape of the upper block was also varied: half of the trials featured a cube as an upper block, while the other half featured an L-shaped block with randomly sampled side lengths.

For the occlusion data set, the variations in the data set were: height of the pillars, height of the occluder, color of the occluder, and color of the moving object (see Figure 9 for more exemplary test sequences). Additionally, in the training data set, the size of the moving object, the width of the pillars, the position of the occluder, and the speed of the moving object were varied.

The variations in the collision event data sets were: stationary object size, moving object size, stationary object color, and moving object color (see Figure 10 for more exemplary test sequences). In the training data set, the camera position and angle were also varied.

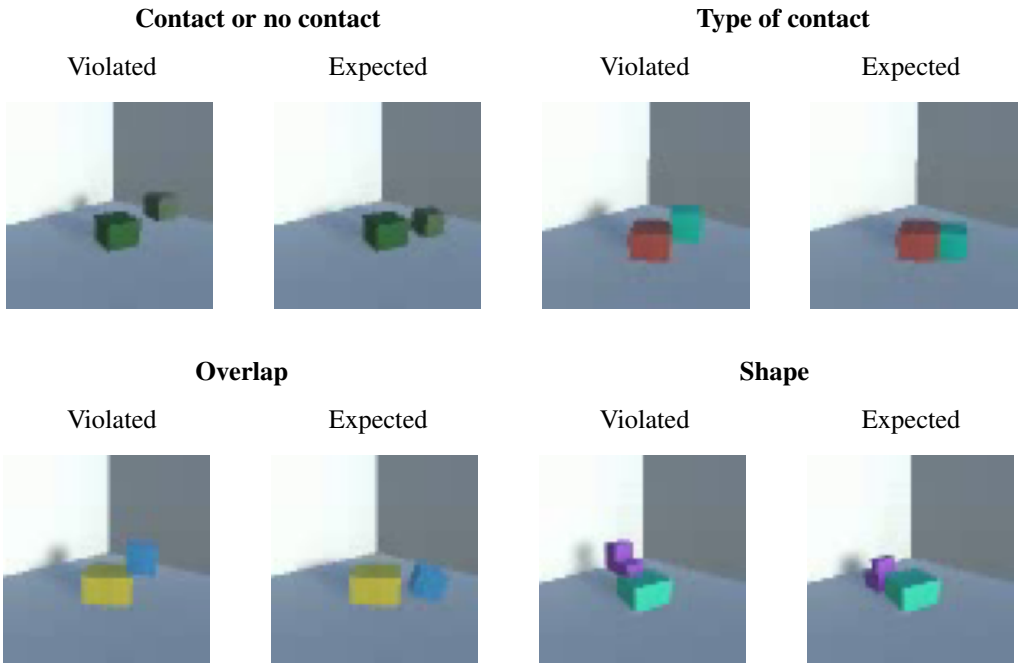

Figure 8. The last frame for example sequences from the support event test data set. From left to bottom and top to bottom, the conditions are *contact or no contact*, *type of contact*, *overlap*, and *shape* with each a violated sequence left and an expected sequence right.

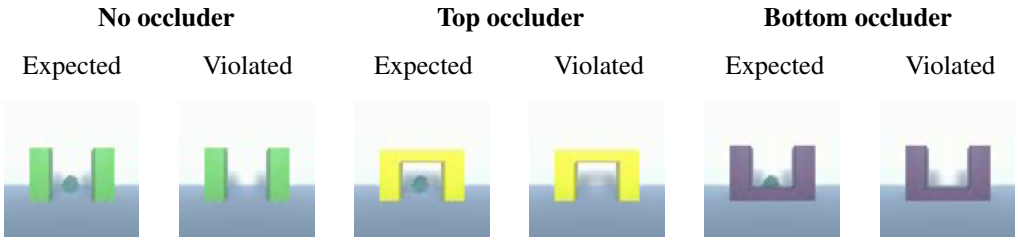

Figure 9. The middle frame for example sequences from the occlusion event data set. From left to right, the conditions are *no occluder*, *top occluder*, and *bottom occluder*. Each condition consists of two sequences: the left sequence shows the expected sequence. The right sequence shows a violation where the moving object only appears on the outside of the occluders.

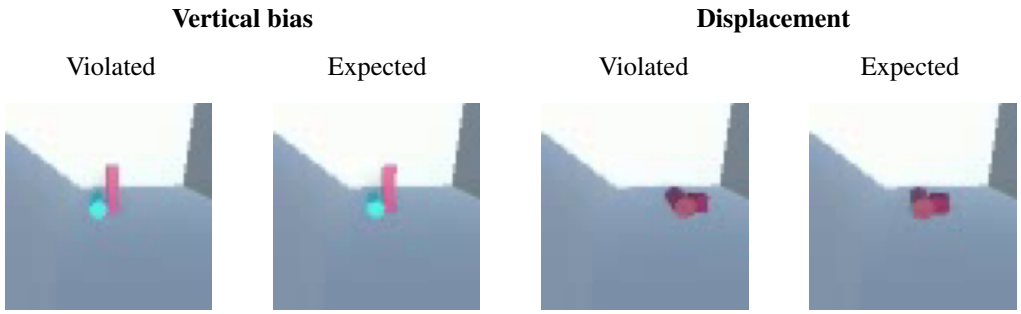

Figure 10. The last frame for example sequences from the collision event data set. From left to right, the conditions are *displacement* and *no vertical bias*. For each condition, there is a violated sequence left and an expected sequence right.

D. Model reconstructions

D.1. Support events

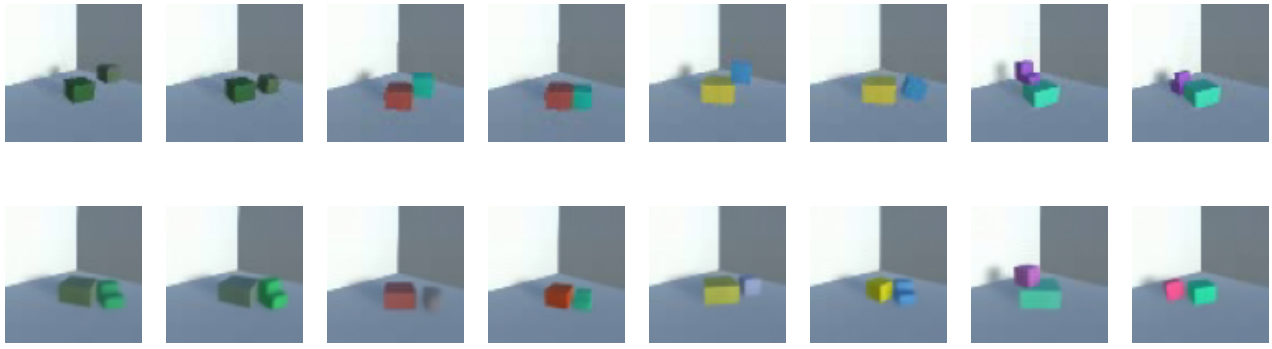

Figure 11. The top row shows the last frame from an example batch of the support event test data set. The bottom row shows the reconstructions by the model with $\beta = 1$ and using open loop reconstruction given only the first frame.

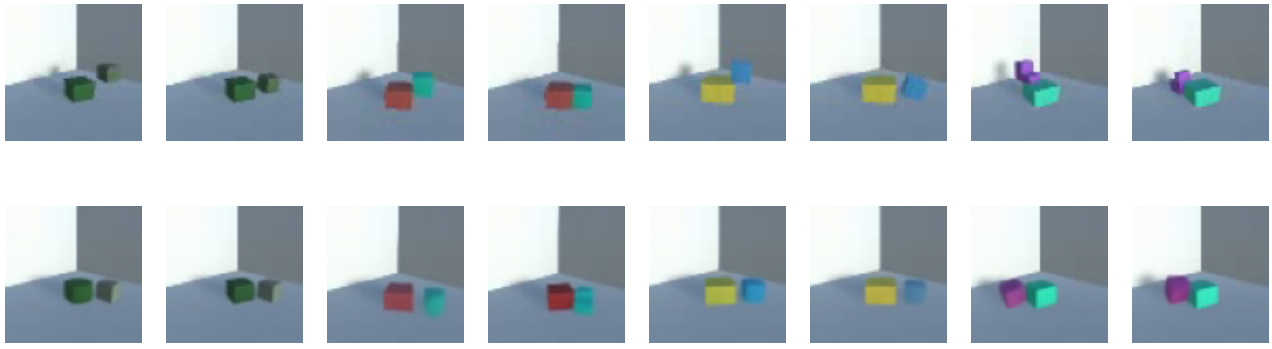

Figure 12. The top row shows the last frame from an example batch of the support event test data set. The bottom row shows the reconstructions by the model with $\beta = 1$ and using open loop reconstruction given the first two frames.

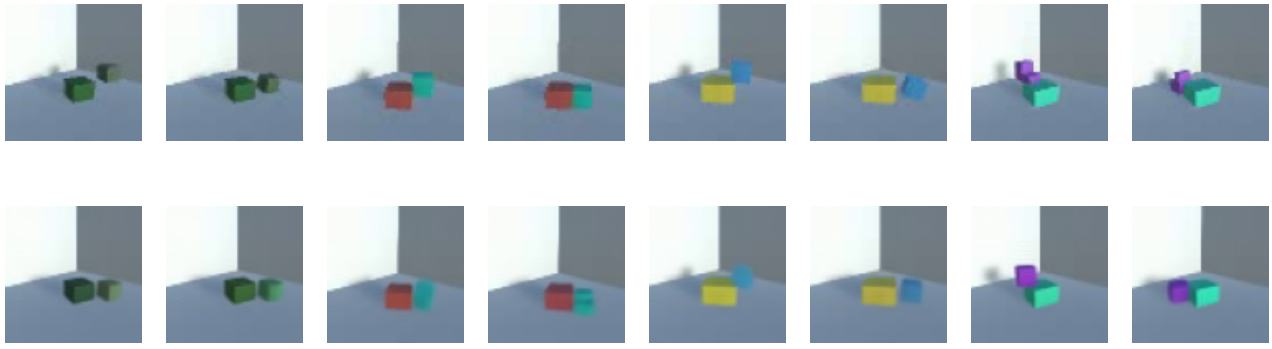

Figure 13. The top row shows the last frame from an example batch of the support event test data set. The bottom row shows the reconstructions by the model with $\beta = 1$ and using closed loop reconstruction.

D.2. Occlusion events

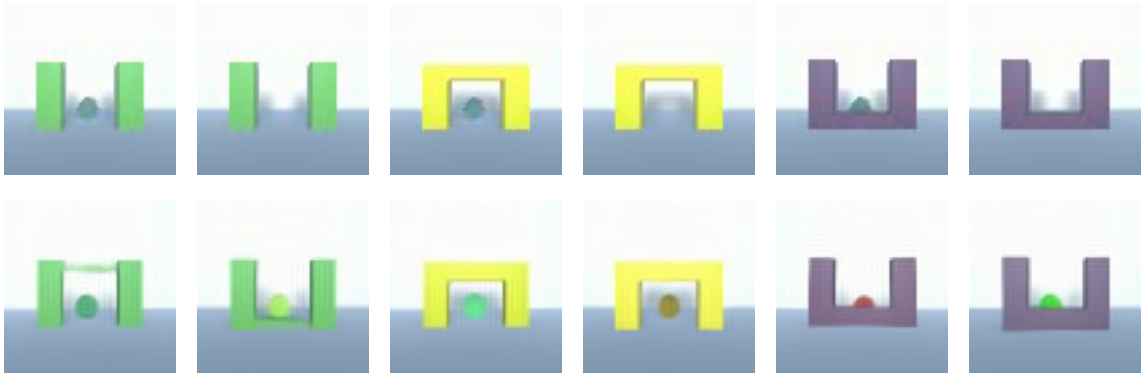

Figure 14. The top row shows the middle frame from an example batch of the occlusion event test data set. The bottom row shows the reconstructions by the model with $\beta = 1$ and using open loop reconstruction given only the first frame.

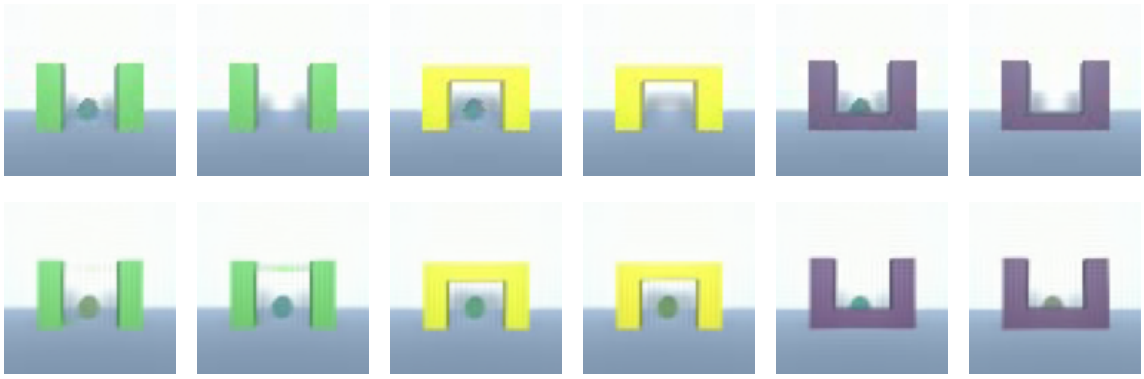

Figure 15. The top row shows the middle frame from an example batch of the occlusion event test data set. The bottom row shows the reconstructions by the model with $\beta = 1$ and using open loop reconstruction given the first two frames.

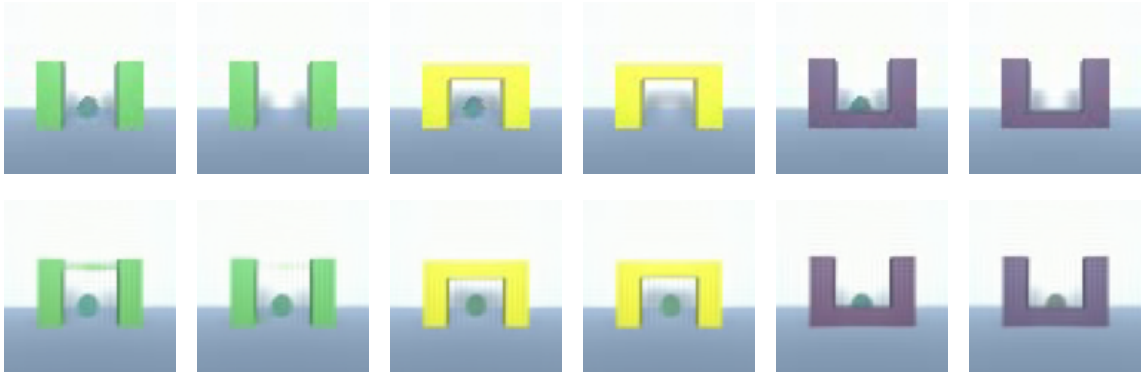

Figure 16. The top row shows the middle frame from an example batch of the occlusion event test data set. The bottom row shows the reconstructions by the model with $\beta = 1$ and using closed loop reconstruction.

D.3. Collision events

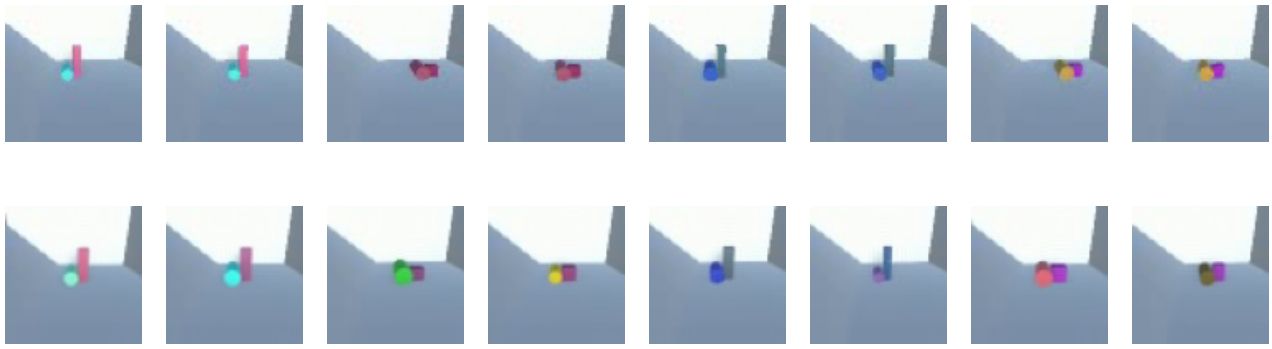

Figure 17. The top row shows the last frame from an example batch of the collision event test data set. The bottom row shows the reconstructions by the model with $\beta = 1$ and using open loop reconstruction given only the first frame.

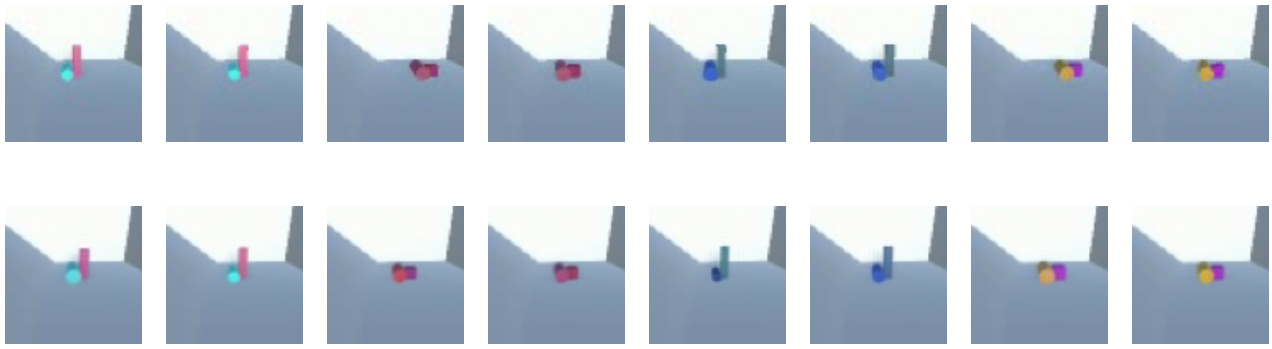

Figure 18. The top row shows the last frame from an example batch of the collision event test data set. The bottom row shows the reconstructions by the model with $\beta = 1$ and using open loop reconstruction given the first two frames.

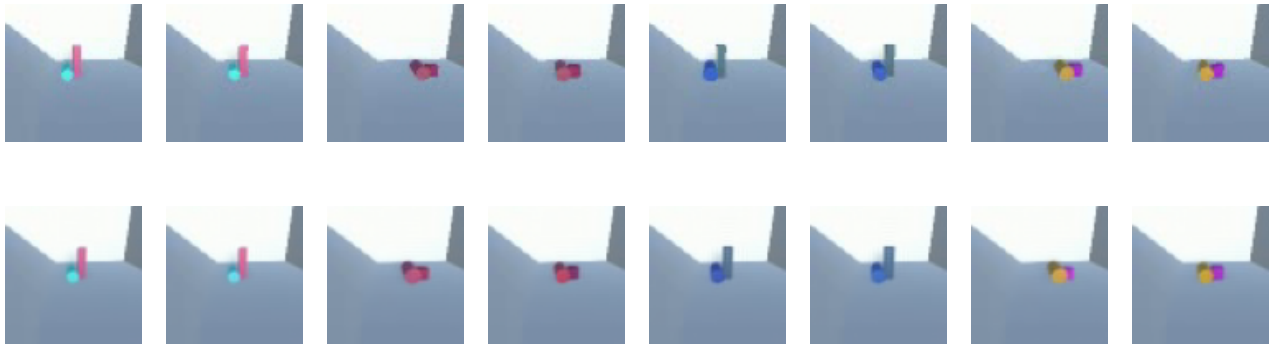

Figure 19. The top row shows the last frame from an example batch of the collision event test data set. The bottom row shows the reconstructions by the model with $\beta = 1$ and using closed loop reconstruction.

E. Reconstruction errors

Figures 20, 21, and 22 show the reconstruction error for the respective event type data sets. The reconstruction error is given by the negative log-likelihood of the reconstructions given example violated sequences. It is displayed as an overlay on top of the violated sequences.

E.1. Support events

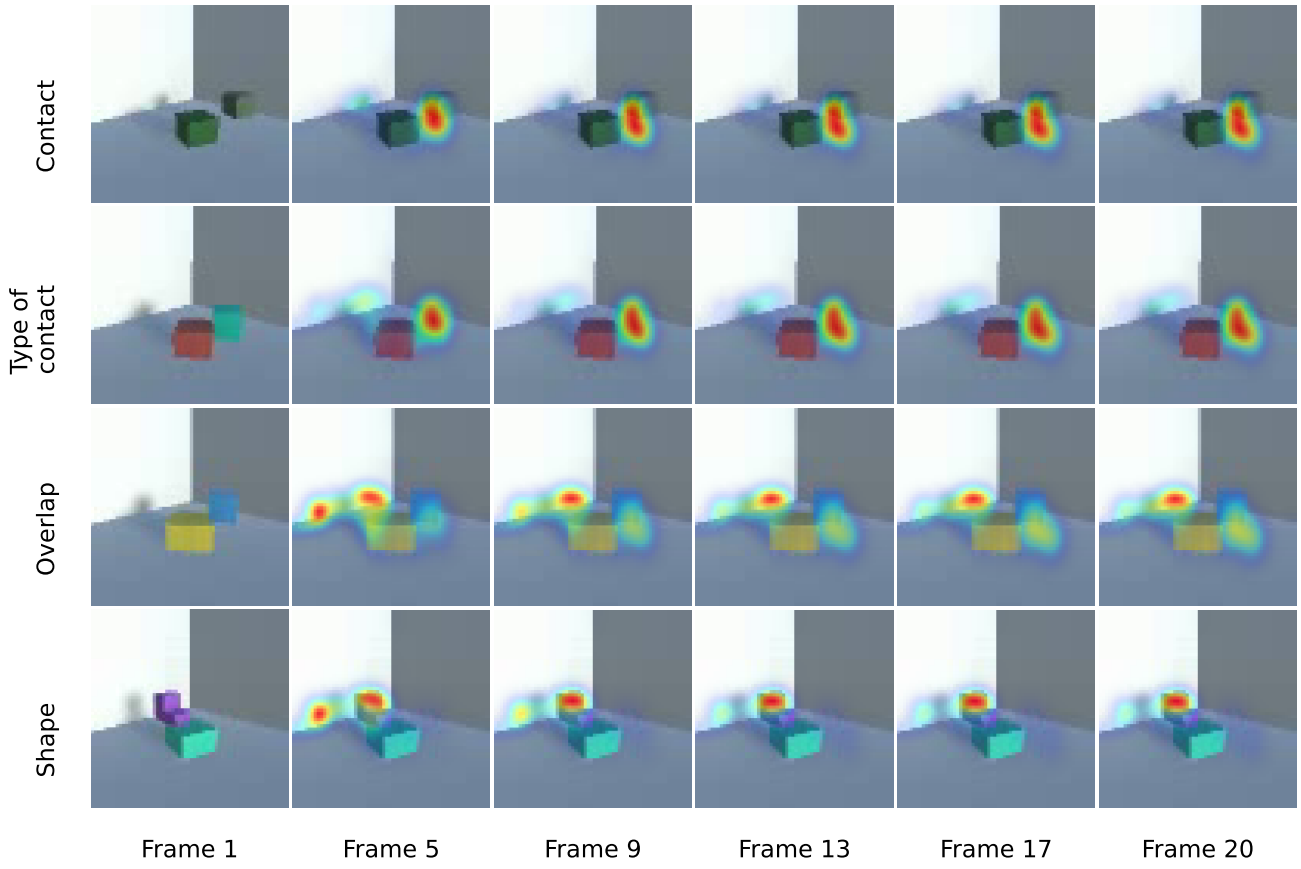

Figure 20. Frames from the violated sequences for one batch of the support event data sets with an overlay showing the negative log-likelihood of the observations under the model with $\beta = 1$ and using open loop reconstruction given the first two frames. We see that the model predominantly focuses on the actual and presumed location of the top cube as well as the shadow of the top cube.

E.2. Occlusion events

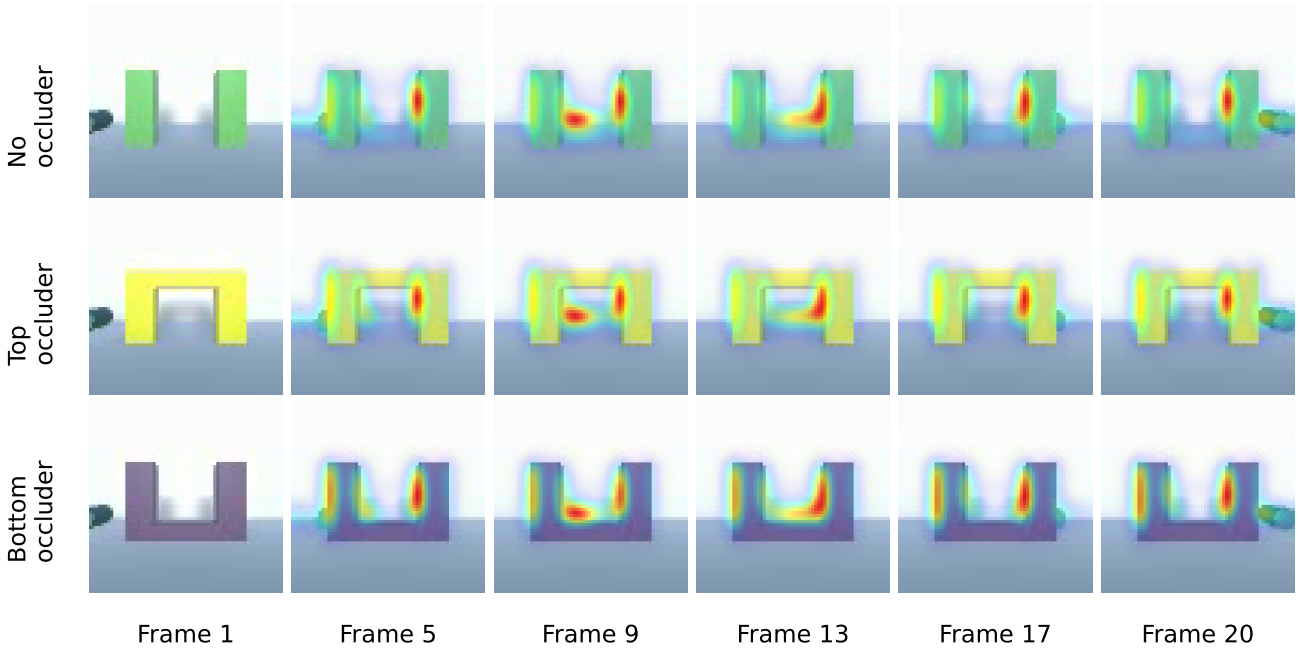

Figure 21. Frames from the violated sequences for one batch of the occlusion event data sets with an overlay showing the negative log-likelihood of the observations under the model with $\beta = 1$ and using open loop reconstruction given the first two frames. The violated sequences show a sphere moving behind an occluder and not reappearing in the gap between the columns. We see that the reconstruction error is large surrounding the edges of the columns. For the bottom occluder sequence, we also see an increased error on the edge of the connection between the two columns.

E.3. Collision events

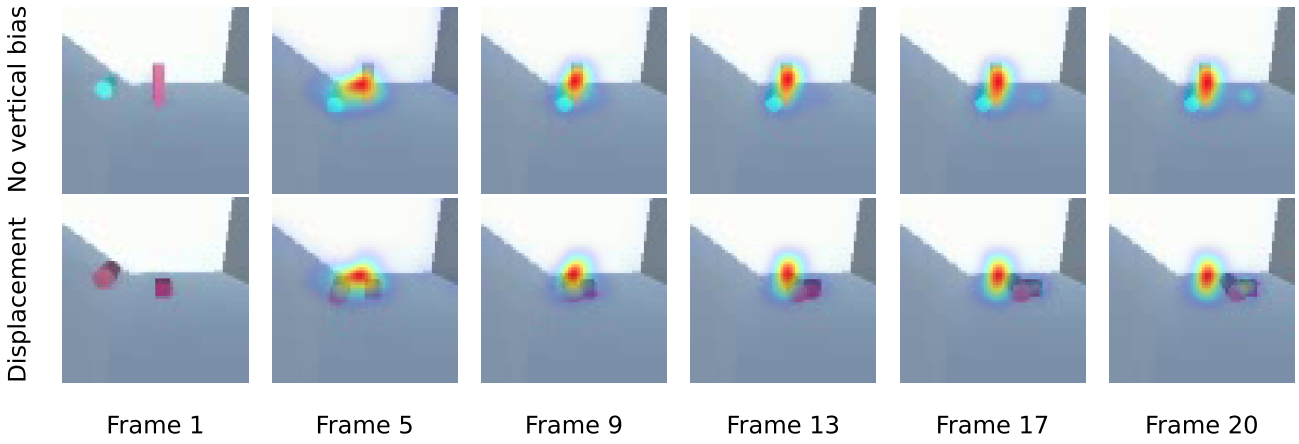

Figure 22. Frames from the violated sequences for one batch of the collision event data sets with an overlay showing the negative log-likelihood of the observations under the model with $\beta = 1$ and using open loop reconstruction given the first two frames. We see that the reconstruction error is especially large at the actual and presumed locations of the stationary object.

F. Closed loop results

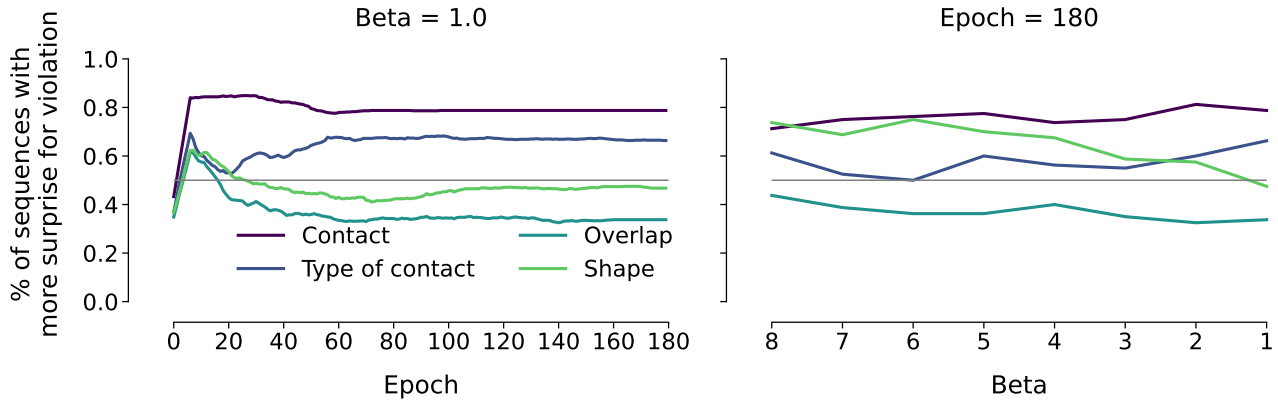

Figure 23. This is the closed loop counterpart to Figure 3. The plot on the left shows the percentage of sequences for which the surprise for the violated sequence exceeds that of the expected sequence for the model with $\beta = 1$ at every epoch and for each condition of the support event data set separately. The lines are smoothed with a uniform kernel of size 10. The plot on the right shows the same metric for fully trained models with different β.

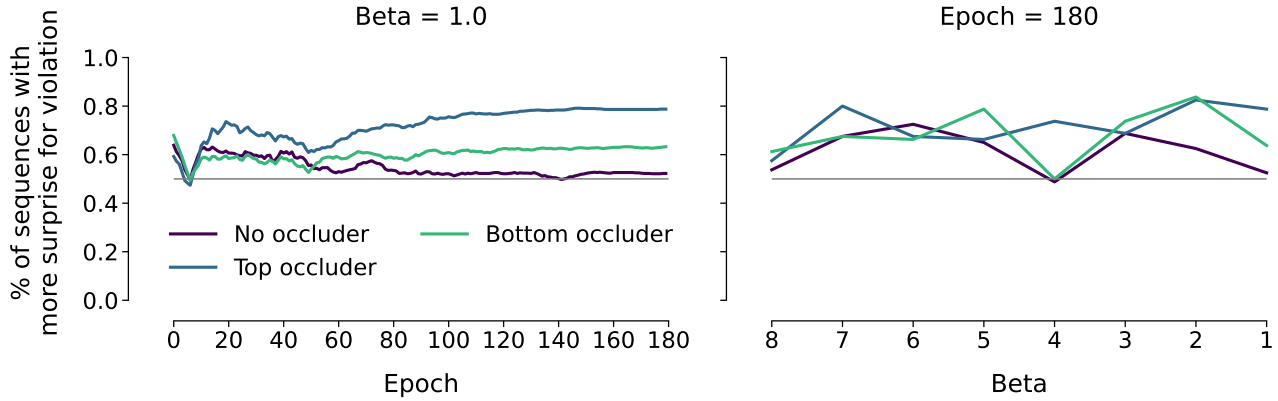

Figure 24. This is the closed loop counterpart to Figure 4. The plot on the left shows the percentage of sequences for which the surprise for the violated sequence exceeds that of the expected sequence for the model with $\beta = 1$ at every epoch and for each condition of the occlusion event data set separately. The lines are smoothed with a uniform kernel of size 10. The plot on the right shows the same metric for fully trained models with different β.

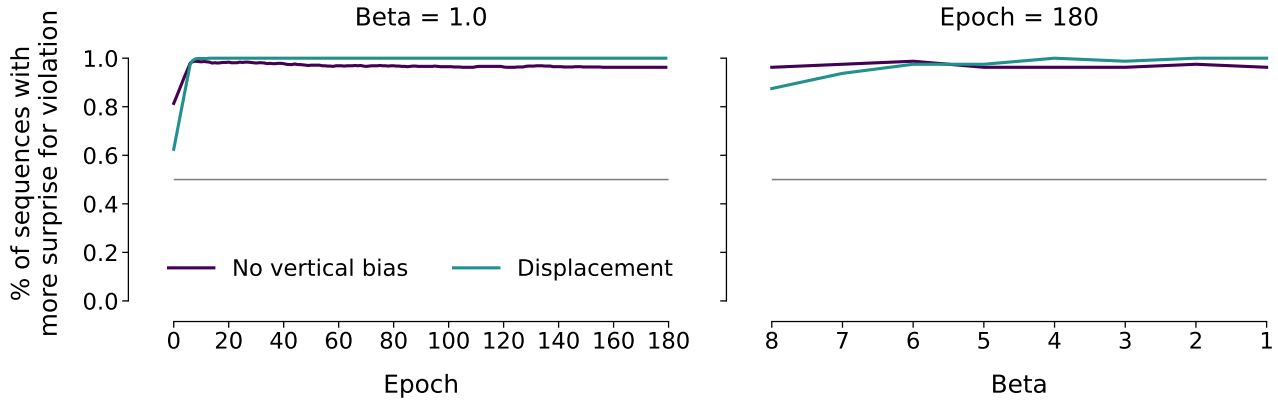

Figure 25. This is the closed loop counterpart to Figure 5. The plot on the left shows the percentage of sequences for which the surprise for the violated sequence exceeds that of the expected sequence for the model with $\beta = 1$ at every epoch and for each condition of the collision event data set separately. The lines are smoothed with a uniform kernel of size 10. The plot on the right shows the same metric for fully trained models with different β.