Understanding Physical Dynamics with Counterfactual World Modeling

Rahul Venkatesh[1*], Honglin Chen[1*], Kevin Feigelis[1*], Daniel M. Bear[1], Khaled Jedoui[1], Klemen Kotar[1], Felix Binder[3], Wanhee Lee[1], Sherry Liu[1], Kevin A. Smith[2], Judith E. Fan[1], and Daniel L. K. Yamins[1]

[1] Stanford [2] MIT [3] UC San Diego

Abstract. The ability to understand physical dynamics is critical for agents to act in the world. Here, we use Counterfactual World Modeling (CWM) to extract vision structures for dynamics understanding. CWM uses a temporally-factored masking policy for masked prediction of video data without annotations. This policy enables highly effective "counterfactual prompting" of the predictor, allowing a spectrum of visual structures to be extracted from a single pre-trained predictor without finetuning on annotated datasets. We demonstrate that these structures are useful for physical dynamics understanding, allowing CWM to achieve the state-of-the-art performance on the Physion benchmark. Project Website: https://neuroailab.github.io/cwm-physics/.

1 Introduction

Physical dynamics understanding involves predicting the effects of physical interactions with objects (e.g. predicting the trajectory of a thrown ball [30], or the direction of a falling stacked block tower [10]). This remains a critical challenge for autonomous agents such as robots and self-driving cars interacting with the world [25]. Existing computer vision algorithms significantly lag behind humans in physical dynamics understanding [12].

One class of existing methods relies on intermediate vision structures such as 2D object segmentations and 3D particle graphs [3,8,9,46,54,55,65,68,69,71,83]. These vision structures are highly useful for accurate dynamics prediction because they abstract away irrelevant details. However, these ground-truth structures are only available in simulated or manually annotated datasets. Scaling these approaches to unlabelled real-world video data remains challenging.

A contrasting class of approaches avoids the use of intermediate structures by learning to predict raw pixels of future video frames [2,4,26,27,37,38,63,81]. While these approaches are directly applicable to real-world videos, learning to predict future frame pixels poses many challenges due to the high-dimensionality of image pixels and the stochasticity of real-world physical dynamics. These unstructured methods substantially underperform approaches with direct access to ground-truth intermediates, especially 3D particles [12].

[*] Equal contribution

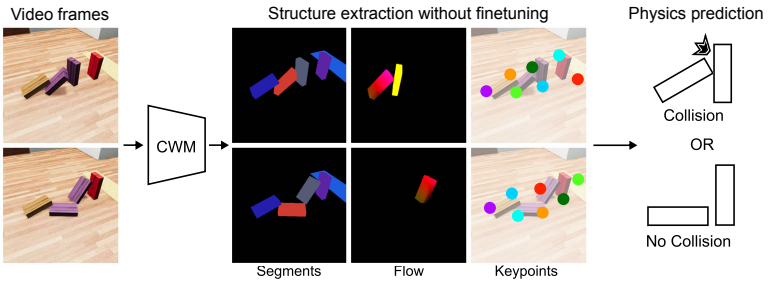


Fig. 1: Overview of the approach. Given an input video of a physical scenario, we extract feature representations and vision structures such as keypoints, optical flow, and segments. These structures are extracted from a single pre-trained CWM predictor without finetuning on annotated datasets. We use the extracted features and structures for dynamics understanding - detecting a past collision or predicting a future collision.

Beyond task-specific methods for physical dynamics prediction, a promising alternative is self-supervised learning of task-agnostic visual representations that transfer well to downstream vision tasks [13,24,40,60,73]. Methods such as DINO [13, 60], masked autoencoder (MAE) [40], and VideoMAE [24, 73, 75] could potentially learn representations useful for dynamics understanding. An additional promise is the emergence of semantic segmentation structure in DINO [13, 60], which could potentially improve dynamics understanding. However, these models are mostly used in a transfer learning or fine-tuning paradigm, which requires annotations. It remains unclear whether they can be prompted to extract meaningful structures without finetuning on annotated datasets.

Therefore, a key research question is designing methods that pre-train on real-world video data without annotations and support extraction of structures for dynamics understanding. In this work, we use a simple and powerful framework, called Counterfactual World Modeling (CWM) [11]. CWM allows extraction of structures useful for understanding dynamics. Figure 1 provides an overview of our approach. We summarize the contributions of CWM below:

(a) We show that using a *temporally-factored masking policy* during pre-training enables powerful prompting abilities. As in VideoMAE, we train a masked predictor on real-world video data. Unlike VideoMAE, in CWM, the predictor only takes in a few patches of the last frame and fully visible preceding frames as inputs, and predicts the remaining patches in the last frame. This temporally-factored masking policy encourages the predictor to concentrate information about transformations between frame pairs into the embeddings of a small number of patch tokens. This in turn enables the predictor to support effective prompting via simple interventions on those few key tokens, allowing the system to answer hypotheticals, such as what will the next frame look like if an object in an image is moved to the right.

(b) We demonstrate that CWM can be prompted to extract multiple vision structures useful for understanding dynamics. As a result of the masking policy,

we can extract structures by feeding CWM different prompts. These structures are extracted from a single predictor without being supervised on annotated datasets. Utilizing the extracted structures, CWM achieves state-of-the-art performance on the challenging Physion benchmark [12].

CWM can be understood in the context of Pearl's Ladder of Causation [32], describing how counterfactual reasoning can be built up from statistical models. The first rung of the Ladder is *Association*, in which a model of the predictable statistical relations between observed events over time is constructed. In CWM, this role is played by the world model itself, the large pretrained predictor which absorbs correlations from observed video inputs. The second rung is *Intervention*, in which at key junctions of the statistical model, observational data are replaced by specific fixed choices ("interventions") intended to produce some desired outcome. In CWM, this role is played by patch-level prompting, whose utility is greatly enabled by the temporally-factored training of the underlying predictor. The third rung of the Ladder is *Counterfactual*, in which the results of interventions are compared to alternative futures to identify true causes of events. In CWM, the comparison between outcomes of counterfactual interventions (prompts) and alternative futures (observed ground truth or observed predictions) are used for structure extraction, which – since they better capture core underlying causes of physical events – end up being useful for improved physical prediction.

In what follows, we review the literature on related works, and describe the core concepts of the CWM framework. We then demonstrate that the extracted structures of CWM are highly useful for physical dynamics understanding. Lastly, we provide an analysis of the quality of the extracted structures and ablation studies of CWM.

2 Related Works

Structured dynamics prediction Researchers have made substantial progress in physical dynamics prediction using structured particle representations as inputs [3, 8, 9, 39, 46, 54, 55, 68, 69, 71]. These approaches simulate large systems of particle-based representations by constructing interaction graphs and propagating information between graph nodes. Besides particle graphs, alternative object structures such as entity locations [83] and keypoints [44] are useful for physical dynamics prediction. However, these methods rely on ground-truth object structures, which are only available in simulated or manually annotated datasets. The scalability of these methods on real-world unlabelled data remains limited.

Video prediction One class of approaches learns physics understanding by predicting the pixels of future video frames [2, 4, 26, 27, 37, 38, 63, 81]. These methods are directly applicable to real-world videos without depending on ground-truth object structures, which are difficult to obtain in general scenarios. Recent video diffusion models [17, 43, 52, 74] and transformer-based prediction models [36, 85] have made progress towards more realistic pixel prediction of future video frames. However, learning to predict future frame pixels poses many challenges due to

the high-dimensionality of image pixels and the stochasticity of real-world physical dynamics. Existing state-of-the-art methods are prone to creating physically implausible motions in the predicted video frames [49].

Self-supervised visual representation learning Beyond task-specific methods for physical dynamics prediction, a promising alternative is self-supervised learning of task-agnostic visual representations from large-scale unlabeled image or video data. These methods learn to generate visual features that transfer well to downstream vision tasks. One school of works leverages different pretext tasks for pre-training [19, 31, 57, 61, 77, 86]. Another class of works models image similarity and dissimilarity between augmented views of an image [13,16,41,58,60,80] and different clips of a video [18,66,88] via constrastive learning. The most recent family of masked visual modeling approaches learns effective visual representations via masking and reconstruction of visual tokens. iGPT [15] and ViT [20] pioneer this direction by training transformers on pixel or patch tokens and exploring masked prediction with patches. MAE [40] introduces autoencoding with an asymmetric encoder-decoder architecture and empirically shows that a high masking ratio is crucial for image tasks. VideoMAE [24, 73] extends to the video domains and shows that an even higher masking ratio leads to strong performance for activity recognition tasks. V-JEPA [7] explores feature prediction as an objective for unsupervised learning from video and achieves state-of-the-art results on activity recognition task in the Something-Something V2 dataset [34]. However, the usefulness of these representations for physical dynamics understanding remains unexplored. Furthermore, these models are mostly used in a transfer learning or fine-tuning paradigm, which requires ground-truth annotations. It remains unclear whether they can be prompted to extract meaningful structures without additional training on annotated data.

3 Method

We discuss in generality the three concepts of CWM by climbing Pearl's Ladder of Causation [32]: (1) temporally-factored masked predictor for learning associations, (2) prompting as interventions and (3) structure extractions using counterfactuals. We will discuss the application of CWM to physical dynamics understanding in Section 4.

3.1 Temporally-factored masked predictor for learning associations

Masked predictor Following MAE [40] and VideoMAE [24, 73], we train an encoder-decoder architecture to reconstruct masked observations of video frames. The input video frames are first divided into non-overlapping spatiotemporal square patches. Then a subset of the patches is masked, and only the remaining visible patches are passed as inputs into the encoder. Finally, the embeded tokens from the encoder and learnable mask tokens, with added positional embedding on all the tokens, are passed as inputs into a shallow decoder to reconstruct the masked patches. The predictor is trained with the mean squared error (MSE)

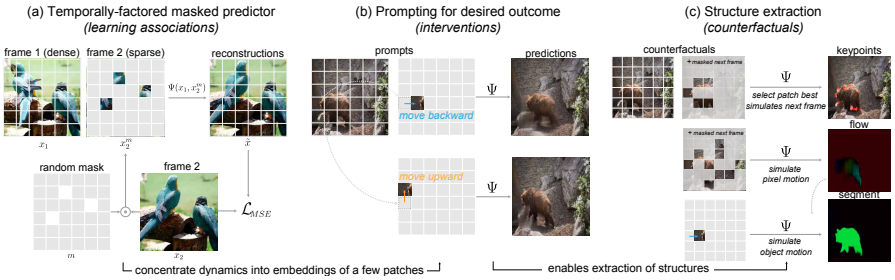


Fig. 2: Climbing the Ladder of Causation with the CWM framework: (a) Temporally-factored masked predictor for association learning. Given a frame pair input, the predictor takes in dense visible patches from the first frame and only a sparse subset of patches from the second frame as inputs, and learns to predict the masked patches. This policy encourages the model to concentrate scene dynamics into embeddings of a few patches. **(b) Prompting as interventations.** As a result of the temporally-factored masking, we can intervene by modifying one or a few visual patches in the prompt and steer the outcome of the predictor. **(c) Structure extraction using counterfactuals.** Multiple vision structures can be extracted by comparing the results of interventions to alternative futures (e.g. observed ground truth or observed predictions).

loss between the reconstructed patches and the original masked patches. The predictor learns the associations between spatiotemporal patches of observed video inputs.

Temporally-factored masking Unlike VideoMAE [24, 73], which randomly samples "tubes" or "cubes" of spatiotemporal patches to be masked, we use a temporally-factored masking policy for video inputs. Without loss of generality, we discuss the masking policy with a frame pair $x_1, x_2 \in \mathbb{R}^{3\times H\times W}$ as input. Given the input frame pair, we train a predictor Ψ:

$$\Psi(x_1^{\alpha}, x_2^{\beta}) = \tilde{x}_2 \tag{1}$$

which takes in first frame x_1 and second frame x_2 with masking ratio $\alpha, \beta \in [0, 1]$. The predictor Ψ predicts the masked patches of x_2, and minimizes the MSE loss between the reconstructed patches $\tilde{x}_2$ and the masked patches of x_2. Figure 2a illustrates this masking policy.

Here, we set the masking ratio α to 0 and β to 0.90, a highly asymmetric masking policy. As a result of this high masking ratio, the predictor Ψ learns to complete the second frame given only a few patches of it, along with the fully visible first frame; the only way it can do this is by inferring scene transformations from a few second-frame patches, then applying these transformations to the first-frame patches to complete the second frame [11]. This implies that the predictor learns to concentrate transformations between frame pairs into the embeddings of a few visible patches. Consequently, modifying the contents of a

few patches, which represent transformations, can exert meaningful control over the next-frame predictions.

3.2 Prompting as interventions

With a pre-trained predictor, at inference time we can replace empirical data observations with interventions intended to produce some desired outcome [32]. As a result of the temporally-factored masking policy, we can modify the original inputs at a few patch locations to generate alternative outcomes using the predictor. To formalize the procedure of intervention, we first define a prompt p as a set of video frames that is given as input to the predictor:

$$p = \{x_1, x_2 \mid x_1, x_2 \in \mathbb{R}^{3\times H\times W}\} \tag{2}$$

where x_2 has a small number of visible patches that specify scene transformations. An intervention $\bar{p}$ is defined as an input to the predictor that has been modified from the initial prompt p. We use two basic types of interventions: (a) appearance prompts, which involve modifications to the first frame x_1, and (b) motion prompts, which involve modifications to the second frame x_2. Given a intervention $\bar{p}$, the associated prediction is the outcome of the predictor $\Psi(\bar{p})$ [11]. Figure 3a shows the predictions for a series of motion prompts. These prompts use a single image, x_1 and construct x_2 by revealing only a few patches in the input image and translating them by a small offset.

3.3 Structure extraction using counterfactuals

The observation of the previous section shows how it is possible to generate counterfactual object motion by modifying the positions of a small number of patches. Next, we discuss how different structure extractions can be specified as counterfactuals [32] by comparing the outcomes of the interventions with alternative futures (e.g. observed ground-truth data or predictions).

Keypoints have been previously defined by manual category-specific annotations [23, 47, 84]. CWM provides a general category-agnostic definition of keypoints as patch locations in x_2 that, when revealed to the predictor, yield the lowest error in the reconstruction $\Psi(p)$ [11]. Let $\mathcal{I}$ be a set of patch locations of an image. The set of keypoints is defined as:

$$\begin{aligned} K(x_1, x_2, n) = &\underset{k\subset\mathcal{I},\,|k|=n}{\arg\min}\ \mathcal{L}(\Psi(p),\, \Psi(\bar{p})) \\ \text{where}\quad &\bar{p} = \{x_1, x_2^m \mid x_2^m \text{ is visible at } k\} \end{aligned} \tag{3}$$

Here, the intervention $\bar{p}$ is the modification of the original input $p = \{x_1, x_2\}$, where the second frame x_2^m is masked everywhere except at keypoint locations. This construction defines a set of dynamical RGB keypoints on x_2. For large values of n, this is in general an intractable optimization problem. In practice, we thus first start with an empty set and add keypoints one at a time to greedily reduce the reconstruction error until n keypoints have been obtained. We show examples of extracted keypoints in the top panel of Figure 3b.

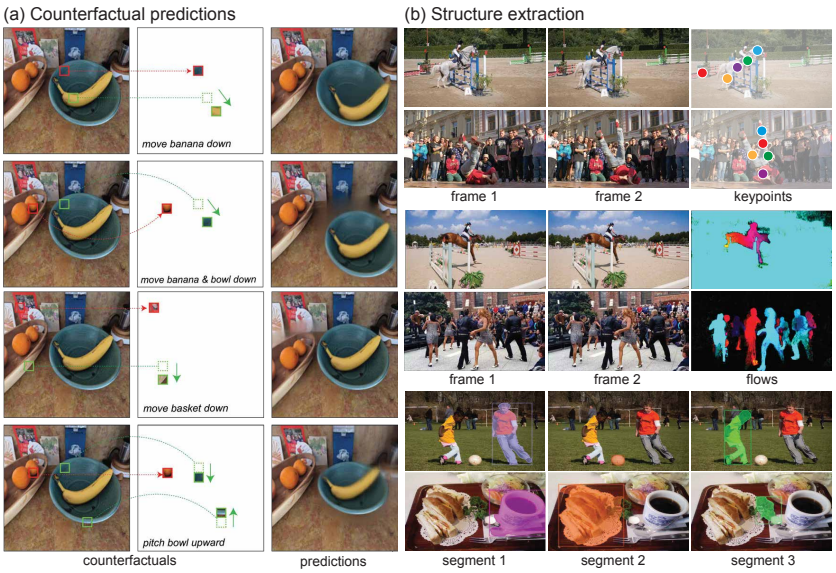


Fig. 3: Counterfactual predictions and structure extraction. **(a) Counterfactual predictions**. A small number of visual patches exert meaningful control of scene dynamics. Each panel shows a prompt consisting of the input image (left), a few patches copied from the input image (middle), and the resulting predictions (right). A red patch is copied into the same location as its source, simulating the appearance of holding an object fixed. A green patch is copied into a different location at an offset from the source location, simulating the appearance of an apparent object motion. **(b) Structure extraction** for keypoints, flows, and segments

Optical flow is the task of estimating per-pixel motion between video frames [72]. To estimate per-pixel motion, we introduce an appearance intervention that adds a small perturbation to the pixel in the first frame. We can estimate the pixel motion by localizing the perturbation response in $\Psi(\bar{p})$ [11].

More specifically, given a prompt $p = \{x_1, x_2^\beta\}$ and a pixel location (i, j), we construct an intervention $\bar{p} = \{x_1 + \delta_{ij}, x_2^\beta\}$, which adds a small perturbation δ_{ij} to the first frame at the pixel location. This creates a perturbed first frame by modifying its appearance at a pixel location. For this reason, we call this an appearance intervention. With a perturbed first frame, the predictor propagates the perturbation in the next frame, under the original scene transformations specified by x_2^β. The corresponding pixel location in the next frame can be localized by finding the peak of the perturbation response. The perturbation response in the next frame can be computed as the absolute difference between the counterfactual prediction $\Psi(\bar{p})$ and the observed prediction $\Psi(p)$. Then, we locate the peak of the perturbation response by taking an argmax over the set of patch locations $\mathcal{I}$. The flow at pixel location (i, j) is then defined as the spatial displacement between (i, j) and the peak of perturbation response:

$$F_{i,j}(x_1, x_2) = \arg\max_{\mathcal{I}} |\Psi(\bar{p}) - \Psi(p)| - (i, j) \tag{4}$$

This algorithm is simple and often effective, as shown in the middle panel of Figure 3b, but it might fail in two ways. First, one of the revealed patches in x_2^β may cover the place where the perturbation at location (i, j) is expected to move. This can be remedied by running the above procedure for multiple random choices of x_2^β and taking their average perturbation responses [11].

A second potential failure mode is that the intervention $\bar{p}$ might be out of distribution for Ψ, which could happen when the perturbation δ_{ij} is too large [11]. On the other hand, if the perturbation is too small, it might not be detected and moved accurately. This can be naturally addressed by using infinitesimal perturbations. We normalize the magnitude of the perturbation response by the magnitude of the perturbation as the limit goes to zero. This is exactly the derivative of the Ψ at location (i, j):

$$\lim_{\delta_{ij} \to 0} \frac{|\Psi(\bar{p}) - \Psi(p)|}{|\delta_{ij}|} = \nabla_x \Psi \Big|_{(i,j)} \tag{5}$$

To simultaneously estimate optical flow at all locations of an input frame, we can compute the Jacobian of Ψ. This is a tensorial operation that can be computed once at all pixels using PyTorch autograd [1]. We describe more details about the procedures of extracting flow in the supplementary material.

Segmentation is defined as a collection of physical stuff that moves together under the application of everyday physical actions [14]. This is inspired by the notion of Spelke object in infant object recognition: infants tend to group scene elements that move together as a single object [70]. CWM extracts segmentation of objects by motion interventions, which simulate object motion at a pixel location, followed by grouping parts of the image that move together.

Given a single image x as input, we define an intervention $\bar{p} = \{x, \bar{x}^m\}$. These prompts produces the second frame $\bar{x}^m$ by revealing only a few patches in the input image and translating those patches by a small offset. With a temporally-factored masked predictor, moving a few patches in the prompt will cause the entire object to move in the resultant counterfactual predictions $\Psi(\bar{p})$. Segments can be extracted by thresholding the flows between the input image x and $\Psi(\bar{p})$:

$$S(x) = F(x, \Psi(\bar{p})) > 0 \tag{6}$$

Once a segment is extracted, we iterate the procedure above to refine the segment by revealing more patches within the segment region into $\bar{x}^m$ and translating patches in the same direction. We set the number of iterations as 3. To automatically discover multiple objects in a single image, we iteratively extract segments at pixel locations that are not part of a discovered object. Once an object segment is discovered, we reveal patches that are not within the segment

region and repeat the procedure to discover the next object. We show examples of extracted segments in the bottom panel of Figure 3b. We discuss more details in the supplementary material.

4 Experiments

Section 4.1 first investigates the usefulness of the extracted structures for downstream physical dynamics understanding tasks. Section 4.2 evaluates the quality of counterfactual motion predictions and extracted visual structures on real-world datasets. Section 4.3 discusses ablations studies on the CWM design.

4.1 Physical Dynamics Understanding

Physion benchmark consists of realistic simulations of diverse physical scenarios where objects are manipulated in a variety of configurations to test different types of physical reasoning such as stability, rolling motion, object linkage, etc. We use the latest version of Physion [12], referred to as Physion v1.5[1], which has improved rendering quality and more physically plausible simulations.

In the ideal scenario, we would evaluate CWM on a real-world physics-understanding benchmark, but such benchmarks are not available. Recent works have shown that simulated data can be highly valuable [48, 72, 87]. Physion is a challenging benchmark as it contains diverse physical phenomena, object dynamics and realistic 3D simulations. This makes it a preferable choice when compared to other benchmarks such as ShapeStacks [35] and IntPhys [67] which contain very limited object dynamics, or Phyre [5] which only operates in 2D environments. Existing video models still significantly lag behind human performance on the Physion benchmark [12]. Moreover, the CWM model is trained on real-world videos from Kinetics-400 dataset [45] and tested on Physion, and is thus a strong generalization test.

The benchmark consists of two tasks: (a) *Object contact prediction* (OCP), which tests the model's ability to *predict* whether two objects *will* contact at some point in the future given a context video, and (b) *Object contact detection* (OCD) which tests the model's ability to *detect* if two objects have already come into contact in the observed video. The video stimuli are generated in such a way that the model needs to have an understanding of the physical dynamics in order to answer the contact-related question correctly. Figure 4 shows example stimuli for the two tasks. For both tasks, the two objects of interest are rendered with red and yellow texture to cue the model.

Evaluation protocol We follow the three-step evaluation protocol of the Physion benchmark [12]. First, we extract features from the last layer of a frozen pre-trained encoder on a training set of 5,600 videos for OCP and OCD tasks, respectively. For image-based methods, features are extracted from 4 frames that are 150 ms apart. For video-based methods, the input frames are fed to the

[1] https://physion-benchmark.github.io

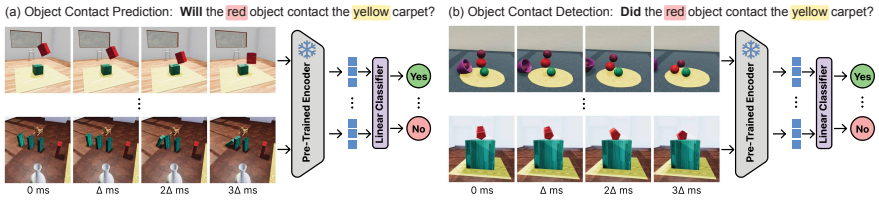


Fig. 4: Physion v1.5 evaluation protocol. We evaluate on two physical dynamics understanding tasks – **(a) Object contact prediction** where the model is asked to predict contact events in the future and **(b) Object contact detection** where the model is asked to reason about contact events that occur in the observed video stimulus. The objects of interest for which we want to ask the contact question are rendered with red and yellow texture to cue the model.

model at the specific frame rate used during their training. Second, the extracted features are used to train a logistic regression model to predict the contact label for the given video stimulus. Lastly, the trained classifier is evaluated on a test set of 1,000 videos across different physical scenarios.

Baseline methods We compare CWM with five classes of baseline approaches: (a) video prediction models including MCVD [74], R3M [56], FitVid [4], and TECO [82], (b) self-supervised representation learning methods on images including DINO [13], DINOv2 [60], and MAE [40], (c) self-supervised representation learning methods on videos including VideoMAE [73], VideoMAEv2 [75] and the recent state-of-the-art method V-JEPA [7] (d) vision-language models like GPT4-V [59] and lastly (e) ground truth 3D particle-based dynamics prediction models such as SGNN [39].

Results In Table 1 we report results on the two Physion tasks for both CWM with ViT-B and ViT-L architectures and other baseline methods discussed above. We evaluate CWM with both features and extracted vision structures input to the linear classifier. We find that video prediction models (such as MCVD [74] and TECO [82]) perform poorly especially on the Physion tasks. Self-supervised image representation models on the other hand, are better but they saturate around 72% and 87% for OCP and OCD respectively with the ViT-B architecture. It is interesting to note that CWM outperforms methods such as DINOv2 ViT-g and MAE ViT-H which have 13 and 7 times more parameters. When scaled up to ViT-L, CWM achieves superior performance on OCP.

We find that CWM exhibits superior performance compared to both VideoMAE [73] and VideoMAEv2 [75]. To ensure a fair evaluation, we train a variant of VideoMAE, denoted as VideoMAE*, that matches CWM in terms of the number of frames and patch size, and include comparable structure extractions from the model for linear probing. Our findings indicate that CWM performs better than VideoMAE*. Furthermore, CWM surpasses the recently released V-JEPA [7], a state-of-the-art model for video representation, despite being trained on a considerably smaller dataset. Furthermore, we find on OCP, CWM achieves a performance that closely approaches that of ground truth 3D particle-based

Table 1: State-of-the-art accuracy on Physion v1.5. We compare CWM to five classes of baseline methods across different architectures on the OCP and OCD tasks. We evaluate CWM with both features and extracted structures and find that it achieves state-of-the-art performance on these tasks. Original VideoMAE [73] uses 16 input frames and a patch size of 16. We trained VideoMAE* with 3 input frames, a patch size of 8, and include extracted vision structures from the model for a strictly fair comparison with CWM.

method	training data	arch	param	OCP ↑	OCD ↑
supervised ground truth 3D particle-based model					
SGNN	Physion v1.5	GNN	23 M	76.4	98.8
video prediction models					
MCVD [74]	K400+Ego4D	UNet	251 M	63.4	80.8
R3M [56]	K400+Ego4D	Res50	38 M	67.6	78.1
FitVid [4]	K400+Ego4D	VAE	303 M	64.3	59.5
TECO [82]	K600	vq-gan	160 M	69.3	80.9
self-supervised image representation models					
DINO [13]	IN-1K	ViT-B	86M	72.1	85.4
DINOv2 [60]	LVD-142M	ViT-B	86 M	72.2	87.1
DINOv2 [60]	LVD-142M	ViT-L	304 M	72.2	85.5
DINOv2 [60]	LVD-142M	ViT-g	1.1 B	72.7	84.6
MAE [40]	IN-1K	ViT-B	86 M	72.6	81.6
MAE [40]	IN-1K	ViT-L	304 M	71.6	82.3
MAE [40]	IN-1K	ViT-H	632 M	73.3	80.8
MAE [40]	IN-4.5M	ViT-B	86 M	72.1	81.7
MAE [40]	IN-4.5M	ViT-L	304 M	72.6	81.9
self-supervised video representation models					
VideoMAE [73]	K400	ViT-B	86 M	72.1	85.7
VideoMAE*	K400	ViT-B	86 M	73.2	86.2
VideoMAE [73]	K400	ViT-L	304 M	73.6	86.1
VideoMAE [73]	K400	ViT-H	632 M	73.5	87.5
VideoMAEv2 [75]	U-Hybrid	ViT-g	1.1B	72.2	85.0
V-JEPA [7]	VideoMix2M	ViT-L	304M	73.4	87.0
vision-language models					
GPT4-V [59]	-	-	-	52.9	54.7
CWM	K400	ViT-B	86 M	75.9	**89.1**
CWM	K400	ViT-L	304 M	**76.1**	88.7

simulation models (i.e SGNN [39]) learned on Physion, despite being trained on Kinetics-400 [45] – a considerably different real world dataset.

We also evaluate GPT4-V [59] on Physion v1.5 tasks by providing it with a single composite image with a sequence of four video frames sampled at a gap of 150ms. The model is prompted with questions similar to those in Figure 4 (see supplementary for more details about the specific prompts used). We find GPT4-V scores nearly at chance on OCP and slightly above chance on OCD, which highlights a considerable limitation in the ability of large-scale vision-language models to understand physical scene dynamics.

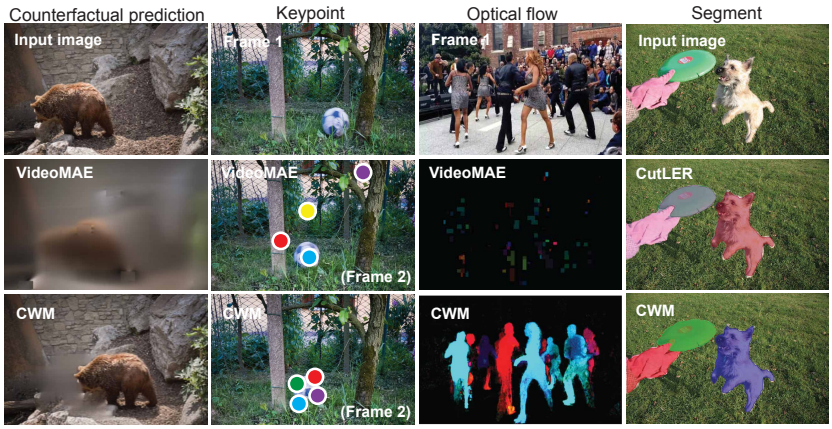


Fig. 5: Qualitative comparison of counterfactual motion prediction and structure extraction on real-world datasets. We find that when we apply our extraction procedures described in Section. 3.3 on VideoMAE, the model fails to generate counterfactual motion and extracts less meaningful structures than CWM. Segments cannot be extracted from VideoMAE due to the failure of counterfactual predictions, and hence not shown in this comparison. This shows the importance of the temporally-factored masking policy during pre-training.

4.2 Analysis of the extracted structures

We analyze the quality of structures extracted by CWM. Although not all baseline methods can perform counterfactual predictions or structure extractions, we apply our procedures to the baseline methods for a fair comparison with CWM. We show CWM yields more meaningful predictions, keypoints, and flows than VideoMAE, enabled by the temporally-factored masking policy. We also show that the quality of segments extracted by CWM is close to the state-of-the-art method CutLER [79], which extracts segments from DINO [13].

Counterfactual prediction We compare CWM and VideoMAE on the quality of counterfactual predictions in Table 2a and Figure 5. We generate counterfactual motions using input images from the DAVIS dataset [64]. The quality of generation is measured by the Fréchet Inception Distance (FID) [42]. CWM significantly outperforms VideoMAE. For a strictly fair comparison, we train another VideoMAE model (referred to as VideoMAE*) with the same number of frames and patch size as CWM. Although the model achieves a slightly lower FID relative to VideoMAE, the reconstructions are still quite blurry without accurate object motions. This illustrates the importance of temporally-factored masking in generating plausible counterfactual predictions.

Keypoints Existing keypoint datasets are generally created with manually specified templates for certain object categories [23, 47, 84]. Therefore, these datasets do not provide suitable quantitative evaluations of CWM keypoint, which are category-agnostic. Figure 5 shows that CWM can extract more meaningful dynamic keypoints as compared to VideoMAE.

Table 2: Quantitative comparison of counterfactual motions, flow and segment extraction on real-world datasets. In (a) we compare to VideoMAE on counterfactual motions and flow. For a strictly fair comparison, we also evaluate VideoMAE* which we trained with the same patch size and number of frames as CWM. In (b) we compare CWM to CutLER [79], which extracts segmentations from DINO [13], and FreeSOLO [78] on the quality of segmentations.

(a) Counterfactual motion (CM) and Flow

Methods	CM (FID ↓)	Flow (F1 ↓)
VideoMAE [73]	213.4	56.3
VideoMAE*	166.3	54.9
CWM	**25.4**	**46.8**

(b) Segments extraction

Methods	Segment (AP ↑)
FreeSOLO [78]	4.3
CutLER [79]	**8.4**
CWM	8.2

Optical flow We evaluate the quality of optical flows on the SPRING benchmark [53] using the F1 metric [29]. We find that CWM is better compared to both VideoMAE and VideoMAE* (See Table. 2a). This is also supported by the qualitative results shown in Figure. 5. We include more qualitative comparisons and additional implementation details in the supplementary.

Segments We extract segments on images from COCO `train2017` [47] using CWM. We follow the same procedures in CutLER [79] to learn a detector using the extracted segments as self-supervision. We train CutLER on COCO training images for a fair comparison. We compare CWM with FreeSOLO [78] and CutLER [79] in Table 2b and Figure 5. CWM outperforms FreeSOLO [78] significantly and attains similar performance to the current state-of-the-art approach CutLER [79]. Although Spelke objects are segment-like structures, the definition of Spelke objects is not exactly aligned with the definition of instance segmentations in the COCO datasets.

4.3 Ablation studies

We ablate the CWM design with the default backbone of ViT-B. Each ablated model is trained for 800 epochs on the Kinetics-400 dataset. Results of the ablation study are reported in Table. 3.

Vision structures We study the importance of each visual structure in understanding dynamics. Adding patch features at keypoint locations improves the OCP accuracy from 73.6% to 74.4%. Enriching these patch features with optical flow patches further improves the accuracy to 75.5%. Finally, including segments achieves a score of 75.9%.

Training schedule We find that a model trained with a longer training schedule of 1600 epochs achieves an OCP score of 75.9% – a relatively small improvement over an 800 epoch trained model (75.4%).

Masking Policy We study the importance of temporal factoring by training a model with a random tube masking strategy, which was originally proposed in VideoMAE [73]. The temporally-factored mask policy is essential for extraction of meaningful vision structures, improving the OCP accuracy improves from 73.2% with tube masking to 75.9% with temporally-factored masking.

Table 3: CWM ablation studies. The best setting is shown in the first row. We investigate the importance of different vision structures, masking policy, training epochs, masking ratio, context frames and patch size.

Ablations	temporal factoring	Input to the classifier				mask ratio	patch size	context frames	training epochs	Metrics	
		feat.	keyp.	flow	segm.					OCP ↑	OCD ↑
Best setting	✓	✓	✓	✓	✓	0.90	8	2	1600	**75.9**	**89.1**
Structures	✓	✓	✓	✓	✗	0.90	8	2	1600	75.5	88.5
	✓	✓	✓	✗	✗	0.90	8	2	1600	74.4	89.1
	✓	✓	✗	✗	✗	0.90	8	2	1600	73.6	89.1
Training epochs	✓	✓	✓	✓	✓	0.90	8	2	800	75.4	88.9
Masking policy	✗	✓	✓	✓	✓	0.90	8	2	800	73.2	86.2
Masking ratio	✓	✓	✓	✓	✓	0.85	8	2	800	75.0	88.9
	✓	✓	✓	✓	✓	0.95	8	2	800	74.6	88.3
	✓	✓	✓	✓	✓	0.99	8	2	800	72.5	86.6
Context frames	✓	✓	✓	✓	✓	0.90	8	1	800	71.0	85.2
	✓	✓	✓	✓	✓	0.90	8	4	800	68.5	79.9
Patch size	✓	✓	✓	✓	✓	0.90	16	2	800	74.2	88.8

Mask ratio We observe that a high ratio on the last frame (90%) during model training achieves good performance on both the OCP and OCD tasks. This trend aligns with our aforementioned hypothesis that the dynamics between frame pairs at a short timescale has a low-dimensional causal structure, which can be concentrated into a small number of tokens.

Context length We compare the performance of CWM with different numbers of context frames. CWM with 2 context frames during pre-training performs better as compared to using 1 context frame. However, including 4 context frames degrades the performance.

Patch size Our analysis indicates that the patch size used for training the model can influence the performance; specifically, a patch size of 8 yields a superior OCP accuracy of 75.9%, compared to a patch size of 16, which results in a lower accuracy of 74.2%.

5 Conclusion

In this work, we show that a simple temporally-factored masking policy during pre-training enables powerful prompting abilities. As a result, we can use counterfactual prompts and their associated predictions to extract vision structures, which abstract away irrelevant details and thus end up being useful for improved dynamics understanding. As compared to random masking, temporally-factored masking policy allows more meaningful and useful structures to be extracted from the pre-trained predictor. CWM achieves state-of-the-art results on the challenging Physion benchmark as compared to previous self-supervised methods, approaching the performance of the best supervised methods in terms of object contact prediction accuracy.

Acknowledgements This work was supported by the following awards: To D.L.K.Y.: Simons Foundation grant 543061, National Science Foundation CAREER grant 1844724, Office of Naval Research grant S5122, ONR MURI 00010802 and ONR MURI S5847. We also thank the Google TPU Research Cloud team for computing support.

References

1. Pytorch autograd, https://pytorch.org/tutorials/beginner/blitz/autograd_tutorial.html, accessed: March 3, 2024
2. Agrawal, P., Nair, A.V., Abbeel, P., Malik, J., Levine, S.: Learning to poke by poking: Experiential learning of intuitive physics. Advances in neural information processing systems **29** (2016)
3. Ajay, A., Bauza, M., Wu, J., Fazeli, N., Tenenbaum, J.B., Rodriguez, A., Kaelbling, L.P.: Combining physical simulators and object-based networks for control. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 3217–3223. IEEE (2019)
4. Babaeizadeh, M., Saffar, M.T., Nair, S., Levine, S., Finn, C., Erhan, D.: Fitvid: Overfitting in pixel-level video prediction. arXiv preprint arXiv:2106.13195 (2021)
5. Bakhtin, A., van der Maaten, L., Johnson, J., Gustafson, L., Girshick, R.: Phyre: A new benchmark for physical reasoning. Advances in Neural Information Processing Systems **32** (2019)
6. Bao, H., Dong, L., Piao, S., Wei, F.: Beit: Bert pre-training of image transformers. arXiv preprint arXiv:2106.08254 (2021)
7. Bardes, A., Garrido, Q., Ponce, J., Rabbat, M., LeCun, Y., Assran, M., Ballas, N.: Revisiting feature prediction for learning visual representations from video. arXiv preprint (2024)
8. Bates, C.J., Yildirim, I., Tenenbaum, J.B., Battaglia, P.: Modeling human intuitions about liquid flow with particle-based simulation. PLoS computational biology **15**(7), e1007210 (2019)
9. 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)
10. Battaglia, P.W., Hamrick, J.B., Tenenbaum, J.B.: Simulation as an engine of physical scene understanding. Proceedings of the National Academy of Sciences **110**(45), 18327–18332 (2013)
11. Bear, D.M., Feigelis, K., Chen, H., Lee, W., Venkatesh, R., Kotar, K., Durango, A., Yamins, D.L.: Unifying (machine) vision via counterfactual world modeling. arXiv preprint arXiv:2306.01828 (2023)
12. Bear, D.M., Wang, E., Mrowca, D., Binder, F.J., Tung, H.Y.F., Pramod, R., Holdaway, C., Tao, S., Smith, K., Sun, F.Y., et al.: Physion: Evaluating physical prediction from vision in humans and machines. arXiv preprint arXiv:2106.08261 (2021)
13. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the International Conference on Computer Vision (ICCV) (2021)
14. Chen, H., Venkatesh, R., Friedman, Y., Wu, J., Tenenbaum, J.B., Yamins, D.L., Bear, D.M.: Unsupervised segmentation in real-world images via spelke object inference. In: European Conference on Computer Vision. pp. 719–735. Springer (2022)

15. Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., Sutskever, I.: Generative pretraining from pixels. In: International conference on machine learning. pp. 1691–1703. PMLR (2020)
16. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
17. Chen, X., Wang, Y., Zhang, L., Zhuang, S., Ma, X., Yu, J., Wang, Y., Lin, D., Qiao, Y., Liu, Z.: Seine: Short-to-long video diffusion model for generative transition and prediction. arXiv preprint arXiv:2310.20700 (2023)
18. Dave, I., Gupta, R., Rizve, M.N., Shah, M.: Tclr: Temporal contrastive learning for video representation. Computer Vision and Image Understanding **219**, 103406 (2022)
19. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE international conference on computer vision. pp. 1422–1430 (2015)
20. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
21. Ebert, F., Yang, Y., Schmeckpeper, K., Bucher, B., Georgakis, G., Daniilidis, K., Finn, C., Levine, S.: Bridge data: Boosting generalization of robotic skills with cross-domain datasets. arXiv preprint arXiv:2109.13396 (2021)
22. elenium Contributors: Selenium: Browser automation framework, https://www.selenium.dev/
23. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision **88**, 303–338 (2010)
24. Feichtenhofer, C., Li, Y., He, K., et al.: Masked autoencoders as spatiotemporal learners. Advances in neural information processing systems **35**, 35946–35958 (2022)
25. Finn, C., Goodfellow, I., Levine, S.: Unsupervised learning for physical interaction through video prediction. Advances in neural information processing systems **29** (2016)
26. Finn, C., Levine, S.: Deep visual foresight for planning robot motion. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 2786–2793. IEEE (2017)
27. Fragkiadaki, K., Agrawal, P., Levine, S., Malik, J.: Learning visual predictive models of physics for playing billiards. arXiv preprint arXiv:1511.07404 (2015)
28. Gan, C., Schwartz, J., Alter, S., Mrowca, D., Schrimpf, M., Traer, J., De Freitas, J., Kubilius, J., Bhandwaldar, A., Haber, N., et al.: Threedworld: A platform for interactive multi-modal physical simulation. arXiv preprint arXiv:2007.04954 (2020)
29. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013)
30. Gerstenberg, T., Peterson, M.F., Goodman, N.D., Lagnado, D.A., Tenenbaum, J.B.: Eye-tracking causality. Psychological science **28**(12), 1731–1744 (2017)
31. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. arXiv preprint arXiv:1803.07728 (2018)
32. Goldberg, L.R.: The book of why: The new science of cause and effect: by judea pearl and dana mackenzie, basic books (2018). isbn: 978-0465097609. (2019)

33. Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., He, K.: Accurate, large minibatch sgd: Training imagenet in 1 hour. arXiv preprint arXiv:1706.02677 (2017)
34. Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., et al.: The" something something" video database for learning and evaluating visual common sense. In: Proceedings of the IEEE international conference on computer vision. pp. 5842–5850 (2017)
35. Groth, O., Fuchs, F.B., Posner, I., Vedaldi, A.: Shapestacks: Learning vision-based physical intuition for generalised object stacking. In: Proceedings of the european conference on computer vision (eccv). pp. 702–717 (2018)
36. Gupta, A., Tian, S., Zhang, Y., Wu, J., Martín-Martín, R., Fei-Fei, L.: Maskvit: Masked visual pre-training for video prediction. arXiv preprint arXiv:2206.11894 (2022)
37. Hafner, D., Lillicrap, T., Ba, J., Norouzi, M.: Dream to control: Learning behaviors by latent imagination. arXiv preprint arXiv:1912.01603 (2019)
38. Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., Davidson, J.: Learning latent dynamics for planning from pixels. In: International conference on machine learning. pp. 2555–2565. PMLR (2019)
39. Han, J., Huang, W., Ma, H., Li, J., Tenenbaum, J.B., Gan, C.: Learning physical dynamics with subequivariant graph neural networks. In: Thirty-Sixth Conference on Neural Information Processing Systems (2022)
40. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16000–16009 (2022)
41. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020)
42. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
43. Höppe, T., Mehrjou, A., Bauer, S., Nielsen, D., Dittadi, A.: Diffusion models for video prediction and infilling. arXiv preprint arXiv:2206.07696 (2022)
44. Janny, S., Baradel, F., Neverova, N., Nadri, M., Mori, G., Wolf, C.: Filtered-cophy: Unsupervised learning of counterfactual physics in pixel space. arXiv preprint arXiv:2202.00368 (2022)
45. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
46. Li, Y., Wu, J., Tedrake, R., Tenenbaum, J.B., Torralba, A.: Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. arXiv preprint arXiv:1810.01566 (2018)
47. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
48. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9298–9309 (2023)

49. Liu, Y., Zhang, K., Li, Y., Yan, Z., Gao, C., Chen, R., Yuan, Z., Huang, Y., Sun, H., Gao, J., et al.: Sora: A review on background, technology, limitations, and opportunities of large vision models. arXiv preprint arXiv:2402.17177 (2024)
50. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)
51. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
52. Lu, H., Yang, G., Fei, N., Huo, Y., Lu, Z., Luo, P., Ding, M.: Vdt: An empirical study on video diffusion with transformers. arXiv preprint arXiv:2305.13311 (2023)
53. Mehl, L., Schmalfuss, J., Jahedi, A., Nalivayko, Y., Bruhn, A.: Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
54. Mottaghi, R., Bagherinezhad, H., Rastegari, M., Farhadi, A.: Newtonian scene understanding: Unfolding the dynamics of objects in static images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3521–3529 (2016)
55. Mrowca, D., Zhuang, C., Wang, E., Haber, N., Fei-Fei, L.F., Tenenbaum, J., Yamins, D.L.: Flexible neural representation for physics prediction. Advances in neural information processing systems **31** (2018)
56. Nair, S., Rajeswaran, A., Kumar, V., Finn, C., Gupta, A.: R3m: A universal visual representation for robot manipulation. arXiv preprint arXiv:2203.12601 (2022)
57. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: European conference on computer vision. pp. 69–84. Springer (2016)
58. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
59. OpenAI: Gpt-4 for vision (chatgpt with image input) (2023), https://openai.com/, accessed: October 27, 2023
60. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.Y., Xu, H., Sharma, V., Li, S.W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision (2023)
61. Pathak, D., Girshick, R., Dollár, P., Darrell, T., Hariharan, B.: Learning features by watching objects move. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2701–2710 (2017)
62. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: Computer Vision and Pattern Recognition (2016)
63. Piloto, L.S., Weinstein, A., Battaglia, P., Botvinick, M.: Intuitive physics learning in a deep-learning model inspired by developmental psychology. Nature human behaviour **6**(9), 1257–1267 (2022)
64. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbelaez, P., Sorkine-Hornung, A., Gool, L.V.: The 2017 davis challenge on video object segmentation. arXiv preprint arXiv:1704.00675 (2017)
65. Qi, H., Wang, X., Pathak, D., Ma, Y., Malik, J.: Learning long-term visual dynamics with region proposal interaction networks. arXiv preprint arXiv:2008.02265 (2020)

66. Qian, R., Meng, T., Gong, B., Yang, M.H., Wang, H., Belongie, S., Cui, Y.: Spatiotemporal contrastive video representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6964–6974 (2021)
67. Riochet, R., Castro, M.Y., Bernard, M., Lerer, A., Fergus, R., Izard, V., Dupoux, E.: Intphys 2019: A benchmark for visual intuitive physics understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(9), 5016–5025 (2021)
68. Sanchez-Gonzalez, A., Heess, N., Springenberg, J.T., Merel, J., Riedmiller, M., Hadsell, R., Battaglia, P.: Graph networks as learnable physics engines for inference and control. In: International Conference on Machine Learning. pp. 4470–4479. PMLR (2018)
69. Smith, K., Mei, L., Yao, S., Wu, J., Spelke, E., Tenenbaum, J., Ullman, T.: Modeling expectation violation in intuitive physics with coarse probabilistic object representations. Advances in neural information processing systems **32** (2019)
70. Spelke, E.S.: Principles of object perception. Cognitive science **14**(1), 29–56 (1990)
71. Tacchetti, A., Song, H.F., Mediano, P.A., Zambaldi, V., Rabinowitz, N.C., Graepel, T., Botvinick, M., Battaglia, P.W.: Relational forward models for multi-agent learning. arXiv preprint arXiv:1809.11044 (2018)
72. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. pp. 402–419. Springer (2020)
73. Tong, Z., Song, Y., Wang, J., Wang, L.: Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. Advances in neural information processing systems **35**, 10078–10093 (2022)
74. Voleti, V., Jolicoeur-Martineau, A., Pal, C.: Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. Advances in Neural Information Processing Systems **35**, 23371–23385 (2022)
75. Wang, L., Huang, B., Zhao, Z., Tong, Z., He, Y., Wang, Y., Wang, Y., Qiao, Y.: Videomae v2: Scaling video masked autoencoders with dual masking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14549–14560 (2023)
76. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks for action recognition in videos. IEEE transactions on pattern analysis and machine intelligence **41**(11), 2740–2755 (2018)
77. Wang, X., Gupta, A.: Unsupervised learning of visual representations using videos. In: Proceedings of the IEEE international conference on computer vision. pp. 2794–2802 (2015)
78. Wang, X., Yu, Z., De Mello, S., Kautz, J., Anandkumar, A., Shen, C., Alvarez, J.M.: Freesolo: Learning to segment objects without annotations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14176–14186 (2022)
79. Wang, X., Girdhar, R., Yu, S.X., Misra, I.: Cut and learn for unsupervised object detection and instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3124–3134 (2023)
80. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3733–3742 (2018)
81. Wu, Z., Dvornik, N., Greff, K., Kipf, T., Garg, A.: Slotformer: Unsupervised visual dynamics simulation with object-centric models. arXiv preprint arXiv:2210.05861 (2022)

82. Yan, W., Hafner, D., James, S., Abbeel, P.: Temporally consistent transformers for video generation (2023)
83. Ye, Y., Singh, M., Gupta, A., Tulsiani, S.: Compositional video prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10353–10362 (2019)
84. You, Y., Lou, Y., Li, C., Cheng, Z., Li, L., Ma, L., Lu, C., Wang, W.: Keypointnet: A large-scale 3d keypoint dataset aggregated from numerous human annotations. arXiv preprint arXiv:2002.12687 (2020)
85. Yu, L., Cheng, Y., Sohn, K., Lezama, J., Zhang, H., Chang, H., Hauptmann, A.G., Yang, M.H., Hao, Y., Essa, I., et al.: Magvit: Masked generative video transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10459–10469 (2023)
86. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14. pp. 649–666. Springer (2016)
87. Zheng, Y., Harley, A.W., Shen, B., Wetzstein, G., Guibas, L.J.: Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19855–19865 (2023)
88. Zhuang, C., She, T., Andonian, A., Mark, M.S., Yamins, D.: Unsupervised learning from video with deep neural embeddings. In: Proceedings of the ieee/cvf conference on computer vision and pattern recognition. pp. 9563–9572 (2020)

Supplementary Material

1 CWM pre-training

1.1 Architecture details

Figure 6 provides an overview of the predictor architecture. The input video is first divided into non-overlapping spatiotemporal patches of size 8×8. Then a subset of patches is masked, and only the remaining visible patches are passed as inputs into the transformer encoder. We follow the standard ViT architecture. Following MAE [40], each transformer block in the ViT consists of a multi-head self-attention block and an MLP block, both having LayerNorm (LN). The CWM encoder and decoder have different widths, which are matched by a linear projection after the encoder [40]. Finally, the embedded tokens from the encoder and learnable mask tokens are passed as inputs into a shallow decoder to reconstruct the masked patches. Each spatiotemporal patch has a unique sine-cosine positional embedding. Position embeddings are added to both the encoder and decoder inputs. CWM does not use relative position or layer scaling [6, 40].

1.2 Implementation details

While the discussion about CWM has for simplicity of presentation assumed a frame pair as input, for physical prediction problems it is natural to have an additional context frame to allow object initial velocities to be well-defined. More specifically, given 3 consecutive video frames at 150 ms apart during training, we provide full visibility to the first two context frames and only mask the last frame. In common situations where there is no motion in the first two frames, the three-frame model will recover what a two-frame model would have learned. When there is motion, the three-frame model will additionally learn acceleration, which is essential for physical predictions. For extracting keypoint and flow which

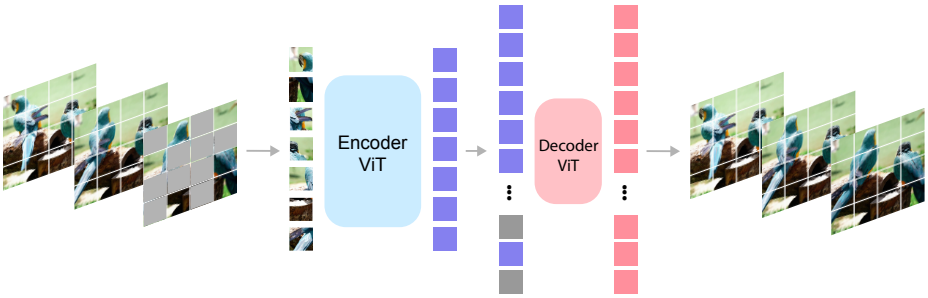


Fig. 6: Architecture of the masked predictor Ψ in the CWM framework.

only require 2 frames as input, we repeat the first frame twice so that the total input length is 3 frames. For extracting segmentation, we are given a single input frame, which we repeat twice and simulate object motions onto the third frame to compute segments.

CWM uses the standard ViT-B and ViT-L architectures with a patch size of 8, which allows structure extraction at a higher resolution. We pre-train CWM on the Kinetics-400 dataset [45], without requiring any specialized sparse operations or temporal downsampling. It takes approximately 6 days to train 1600 epochs on a TPU v4-256 pod.

1.3 Default settings

We show the default pre-training settings in Table 4. CWM does not use color jittering, drop path, or gradient clip. Following ViT's official code, xavier uniform is used to initialize all Transformer blocks. Learnable masked token is initialized as a zero tensor. Following MAE, we use the linear lr scaling rule: $lr = base\_lr \times batch\_size \,/\, 256$ [40].

Table 4: Default pre-training setting of CWM

config	value
optimizer	AdamW [51]
base learning rate	1.5e-4
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$ [15]
accumulative batch size	4096
learning rate schedule	cosine decay [50]
warmup epochs [33]	40
total epochs	1600
flip augmentation	no
augmentation	MultiScaleCrop [76]

2 Structure extraction details and results

In this section, we discuss implementation details of the counterfactual queries for extracting keypoints, optical flow, and segmentations. We also provide more qualitative results of each structure extracted by CWM.

2.1 Keypoint

Implementation details CWM queries keypoints iteratively, starting with an intervention initialized as an initial empty mask and adding visible tokens one-by-one. Note that, whereas the counterfactual queries for optical flow and segmentation involve perturbing the visual input to the predictor Ψ, keypoints arise by varying the prediction model's input mask.

At each iteration, we compute the Mean-Squared-Error (MSE) between the next-frame predictions of Ψ and the ground-truth next frame. We sort the MSE and select the top k locations as candidate keypoints. k is set as 4 by default. For each candidate keypoint, we add its patch content to the intervention and re-compute the MSE between the updated predictions of Ψ and the ground-truth next frame. The candidate keypoint with the minimum MSE error, or equivalently maximum error reduction, is selected as the keypoint output at that iteration. The selected keypoint is added to the intervention and we repeat the procedures above to compute the location of the next keypoint.

Additional qualitative results Figure 7 shows additional qualitative results of the keypoints extracted on DAVIS 2016 [62] and Bridge dataset [21]. We extract 5 keypoints for each example. Our procedure extracts dynamical RGB keypoints in the input frame pairs.

2.2 Optical flow

Implementation details As originally proposed in [11], we simultaneously estimate optical flow at all locations in a frame pair $x_1, x_2 \in \mathbb{R}^{3\times H\times W}$ via the Jacobian of Ψ, denoted as $\mathcal{J}\Psi \in \mathbb{R}^{H\times W\times H\times W}$. The Jacobian of Ψ assigns to element (i, j, k, l) the predictor's change in output at location (k, l) in the second frame due to an infinitessimal change at location (i, j) in the first frame. Flow can be computed using $\mathcal{J}\Psi$ as the following:

$$\textbf{flow}(k,l) = \begin{cases} \textbf{undefined} \text{ (disocclusion)} & \text{if } \textbf{argmax}_{i,j}|\mathcal{J}\Psi(i,j,k,l)| \ll 1 \\ (k,l) - \textbf{argmax}_{i,j}\mathcal{J}\Psi(i,j,k,l) & \text{otherwise} \end{cases} \tag{7}$$

with results averaged over several choices of visible patches in the masked second frame x_2^β. This is a tensorial operation that enables parallel computation for flow at all pixel locations, implemented practically using Jacobian-vector products available in Pytorch autograd. Note that this method detects disocclusion rather than occlusion, since no perturbation at any location in the first frame will cause a response at a point that becomes disoccluded in the second frame.

Additional qualitative results We show additional qualitative results on two distinct datasets: DAVIS 2016 [64] and a recent synthetic dataset SPRING [53]. The results are shown in Figure 8 and 9 respectively.

2.3 Segmentation

Single Spelke object extraction To extract the Spelke object at a pixel location (i, j) of a static image, we first create an intervention that simulates counterfactual motion by taking the patch at location (i, j) and creating a new frame that is largely blank, but in which the content of the patch has been copied (e.g. translated) to a new location $(i+\epsilon_1, j+\epsilon_2)$, where ϵ_1, ϵ_2 are location offsets randomly sampled within a radius $r > 0$. We set r to a fixed fraction (0.2) of the input image size. In addition, we can optionally create the appearance of stopping the counterfactual motion at a location (i', j') by directly copying the patch at that location to the same location in the intervention without offset (or equivalently $r = 0$). Adding the stop-motion patch allows the counterfactual query to isolate a single object, especially in a cluttered scene with multiple objects adjacent to and stacked on top of one another. In practice, different random choices of motion offset and stop-motion patches could potentially yield different counterfactual motion results. We sample 4 different interventions per pixel location, compute flow for the counterfactual motion, average the flow magnitudes of the different samples, and then threshold the mean flow magnitude map at 0.5 to obtain the binary segmentation map at pixel location (i, j). We can use CWM to estimate optical flows, following procedures described in section 2.2. For faster extraction, we use RAFT [72] to estimate the optical flows of different samples.

We also find that the above procedure can be repeated iteratively to refine the segments further. Once a tentative segmentation map is obtained, we can sample more patches within the segment and add them to the intervention to simulate better counterfactual motion. At each iteration, we sample one patch within the segment and add it to the set of patches that simulates counterfactual motion. We additionally sample one patch outside the segment and add it to the set of stop-motion patches that simulates stopping the counterfactual motion. We set the number of iterations to 3 by default.

While the predictor Ψ is trained with input resolution 224×224, Spelke object can be extracted at a different resolution by simply interpolating the position encoding correspondingly. We extract zero-shot segmentations on COCO training images at resolution 480×480 using the ViT-B/8 CWM model.

Multiple Spelke object extraction To automatically discover multiple Spelke objects in a single image, we choose interventions at pixel locations based on a sampling probability distribution α, which has high probability at pixel locations belonging to Spelke objects and low otherwise. Once a segment is discovered, we mask out the probability distribution values using the segments and repeat the process to discover the next object.

We find two choices of sampling distribution work well. One is a movability distribution computed by sampling a few random interventions and averaging the motion responses across multiple predictions. The second choice is the prominence map computed by applying normalized cut to the patch-wise feature similarity matrix as proposed by CutLER [79]. In practice, the second approach yields slightly better qualitative segmentation results. Therefore, we choose the second approach as the default for computing the sampling probability distribution.

Distillation We follow the same procedure in the previous work CutLER [79] to distill segmentations extracted from a large task-agnostic pre-trained model into a smaller instance segmenter for faster and more robust segmentation. The extracted segmentations are used as pseudo annotations to train a downstream instance segmenter in a self-supervised manner.

Additional qualitative results Figure 10a and 10b show more qualitative segmentation results of Spelke objects extracted by CWM on COCO training images, and compare them to those of other baseline methods FreeSOLO [78] and CutLER [79]. In each image, we set the maximum number of Spelke objects to be extracted as 3. Figure 11a and 11b show unsupervised segmentation results from the distilled instance segmenter. We show the results on COCO validation images.

3 Dynamics understanding experiments

3.1 Physion benchmark

Dataset details As discussed in the main text, we use the Physion v1.5 benchmark to evaluate CWM and baseline models on physical dynamics understanding. Physion v1.5 has several key improvements over Physion v1, which are illustrated in Figure 12. More specifically, Physion v1.5 introduces another indoor environment, called the "craft room", in addition to the two environments featured in Physion v1 (see Figure 12d). Furthermore, v1.5 enhances the diversity of lighting conditions by employing a collection of 8 unique HDRI skyboxes, specifically designed to simulate various environmental lighting scenarios. This enhancement allows for dynamic time-of-day simulations in the room by adjusting the orientation of the skyboxes and directional lighting (see Figure 12a). Physion v1.5 also has improved rendering quality and photorealism in comparison to v1 (see Figure 12b and 12c). The physics simulations and rendering are done using the ThreeDWorld simulation platform [28].

Physion v1.5 comprises seven distinct physical scenarios, including collide, drop, dominoes, contain, roll, support, and link. This version comprehensively demonstrates various aspects and challenges of rigid body physics (see Figure 13 for examples of each scenario). We train and test the linear classifier model (as outlined in Section 4.1 of the main text) on all seven scenarios.

Per-scenario OCP and OCD results. In the main text, we presented the OCP and OCD scores averaged across all seven scenarios. Now, we provide a detailed breakdown of performance for each specific scenario, as shown in Table 7 and Table 8.

Qualitative comparisons on OCP and OCD task. We supplement our quantitative results on the Physion v1.5 tasks with qualitative visualizations in Figures 13, 14, 15 and 16. We show several example inputs for each scenario, along with the classification results of a linear probe on top of the CWM model, compared with those of leading baselines in each model category outlined in Table 1 in the main text.

Integrating vision structures for OCP and OCD tasks. In the main text, we demonstrate how zero-shot vision structures extracted from CWM improve OCP and OCD performance on Physion v1.5. This section describes the details of how these structures are integrated prior to linear probing on downstream tasks. Keypoint information is integrated by incorporating patch features at the keypoint locations. Optical flow information is integrated by providing 8×8 patches of optical flow value at the keypoint locations. Finally, segment information is integrated by using the segments to pool the feature map and incorporate the aggregated features for linear probing.

The integration process for segments is detailed as follows: Let $F \in \mathbb{R}^{H \times W \times D}$ be the feature map of an input frame from the last layer of the ViT encoder, where H, W, and D represent height, width, and channel dimension. Suppose we have binary segmentation masks S with dimension $N \times H \times W$, where N denotes the number of masks. We compute the set of aggregated feature vectors for each of the segments,

$$F_{agg} = \left\{ \frac{\sum_{i,j} S_{ij}^n F_{ij}}{\sum_{i,j} S_{ij}^n} \mid 1 \leq n \leq N \right\} \tag{8}$$

where S^n is the n-th segmentation mask and F_{ij} is the feature vector at location (i, j). These aggregated feature vectors, along with the keypoint patch features and flow patches, are concatenated with the original feature map F before being fed into the linear classifier.

3.2 GPT4-Vision prompting

GPT-4V prompting methodology and results. In Figures 17, 18, 19 and 20 we report a few results from testing GPT-4V on the OCP and OCD tasks in Physion v1.5. In each figure, the prompt image is shown on the left and the prompt text along with the GT label and response is shown on the right. To construct the image prompt, we tile four successive video frames (sampled at a frame gap of 150ms) into a 2×2 image with four panels. Each panel contains an RGB frame titled with its timestamp. Following the methodology used for evaluating vision-only models, the objects of interest for the contact-related queries

Table 5: GPT-4V performance on the Physion v1.5 tasks using two different promoting strategies: a) with RGB frames only and b) RGB frames with ground truth segment map overlayed on the objects of interest.

Prompting Method	OCP ↑	OCD ↑
RGB frames	52.9	54.7
RGB frames + GT segment overlay	**58.3**	**67.5**

are rendered with red and yellow textures to provide visual cues for the model. For OCP, the text prompt used was "These are 4 images taken sequentially from a video. If the video were to continue, would the red object touch the yellow surface? Explain your thinking and end with True or False only". For OCD, the prompt used was "These are 4 images taken sequentially from a video. Does the red object touch the yellow surface at any point in the video? Explain your thinking and end with True or False only". GPT4-V achieves 52.9% accuracy for OCP and 54.7% accuracy for OCD. For OCP, the model predicted "contact" 78.7% of the time, while for OCD the model predicted "no contact" 81.5% of the time.

Alternate querying methods. Additionally, we experiment with an alternative querying method to explore the limit of GPT4-V in dynamics understanding. In addition to rendering the objects of interest in red and yellow texture, we apply a bright red and yellow ground-truth segmentation overlay on them to focus the model's attention on these objects (see Figure 17 for visualizations). As shown in Table 5, we find that this querying strategy improves the OCP from 52.9% to 58.3% and the OCD from 54.7% to 67.5%. In the main text, we have also demonstrated a parallel phenomenon with CWM, where integrating segment information led to enhanced performance in related tasks (Table 3 in main text). These observations highlight the importance of vision structures such as segmentation in downstream tasks associated with physical dynamics understanding.

Implementation details. We employ four GPT-4V accounts and retrieve the results using Selenium [22], a browser automation tool. Our methodology adheres to the code framework available at[2].

4 Evaluating CWM on additional benchmarks

4.1 Activity Recognition

We evaluate CWM on the Something-Something V2 benchmark for activity recognition and report the results in Table 6a. To obtain model predictions we train an attentive probe on the feature representation similar to the setup used in

[2] https://github.com/Michelangelo27/chatgpt_selenium_automation

Table 6: Evaluation on additional benchmarks. (a) Activity recognition on Something-Something V2, (b) IntPhys intuitive physics benchmark.

Model	frames	accuracy ↑
VideoMAE	16	**54.7**
VideoMAE*	3	51.3
CWM	3	54.2

(a) Activity recognition on SSv2

Method	B1↓	B2↓	B3↓
VideoMAE	0.40	0.23	0.30
VideoMAE*	0.46	0.30	0.30
VideoMAEv2	0.36	0.30	0.36
CWM	**0.36**	**0.20**	**0.26**

(b) IntPhys

V-JEPA [7]. We find that CWM outperforms VideoMAE trained with the same number of input frames (i.e VideoMAE*) by a fair margin and is comparable to the standard VideoMAE trained on a longer context window of 16 frames.

4.2 IntPhys

In the main text we show evaluations on Physion as it is by far the most challenging and comprehensive benchmark in the literature, consisting of realistic 3D simulations from diverse physical scenarios. Here, we evaluate on IntPhys [67] which is a complementary benchmark to Physion with photorealistic simulations of various intuitive physics tasks. Table 6b reports the relative error metric in IntPhys [67] evaluation on the validation split. For all models, we use the future frame reconstruction error to obtain a plausibility score for a given video. The evaluation comprises of three tasks: B1 tests for object permanence, B2 tests for shape constancy, and B3 tests for spatio-temporal continuity. We find that CWM, despite trained with less number of frames, outperforms VideoMAE. CWM (86M) outperforms VideoMAEv2 (1.1B) despite having fewer parameters. For a fair comparison, we further evaluate VideoMAE* which is trained on the same number of frames as CWM but with tube masking instead of temporally factored masking. The performance gain from CWM further validates the benefit of the temporally-factored masking policy.

Fig. 7: Keypoints extracted by CWM. We extract 5 keypoints for each example on DAVIS 2016 (left) and Brige dataset (right). 23

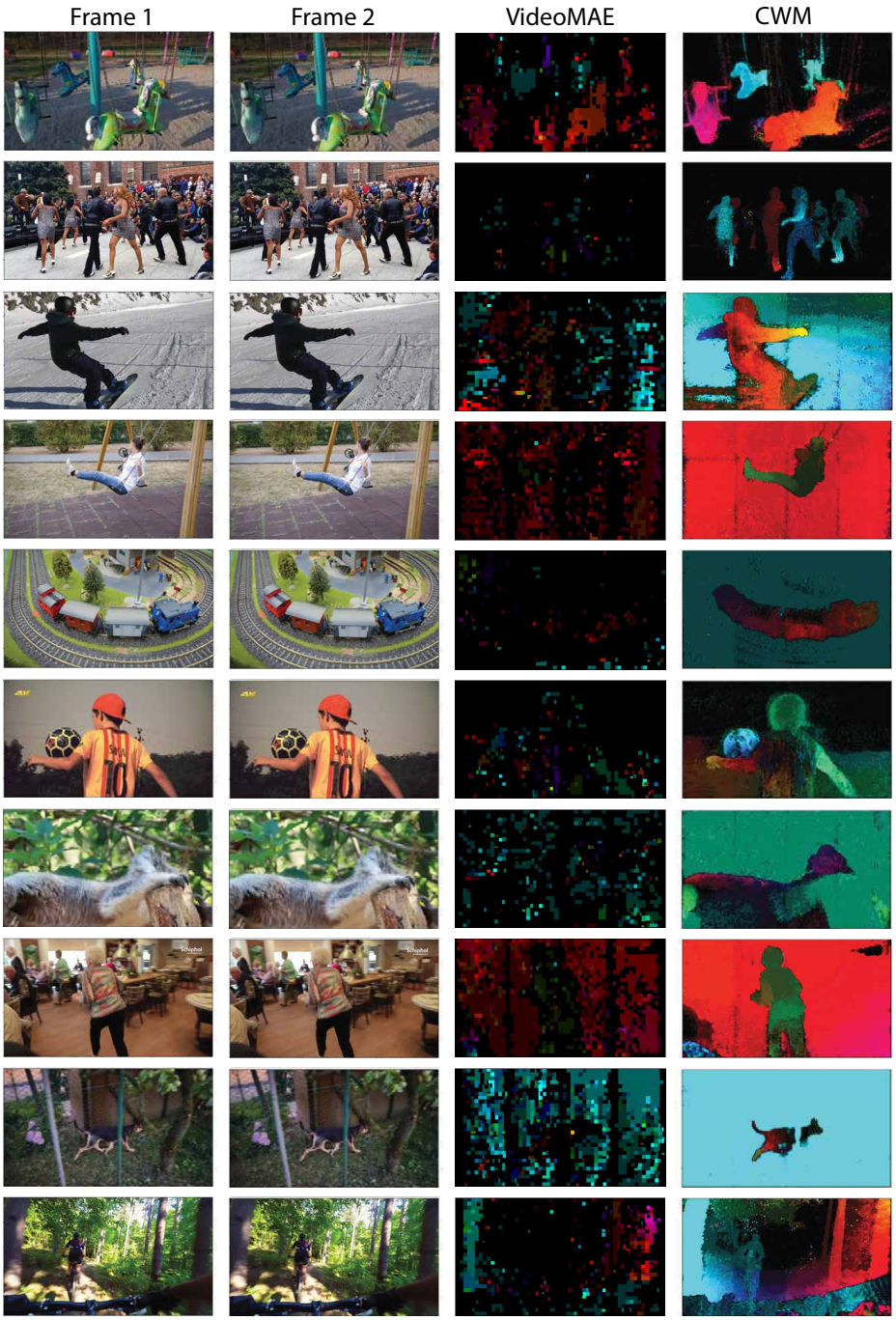


Fig. 8: Additional Optical Flows extracted on the DAVIS dataset. We apply our flow extraction procedure to both CWM and VideoMAE predictors, and compare the extracted flows. 23

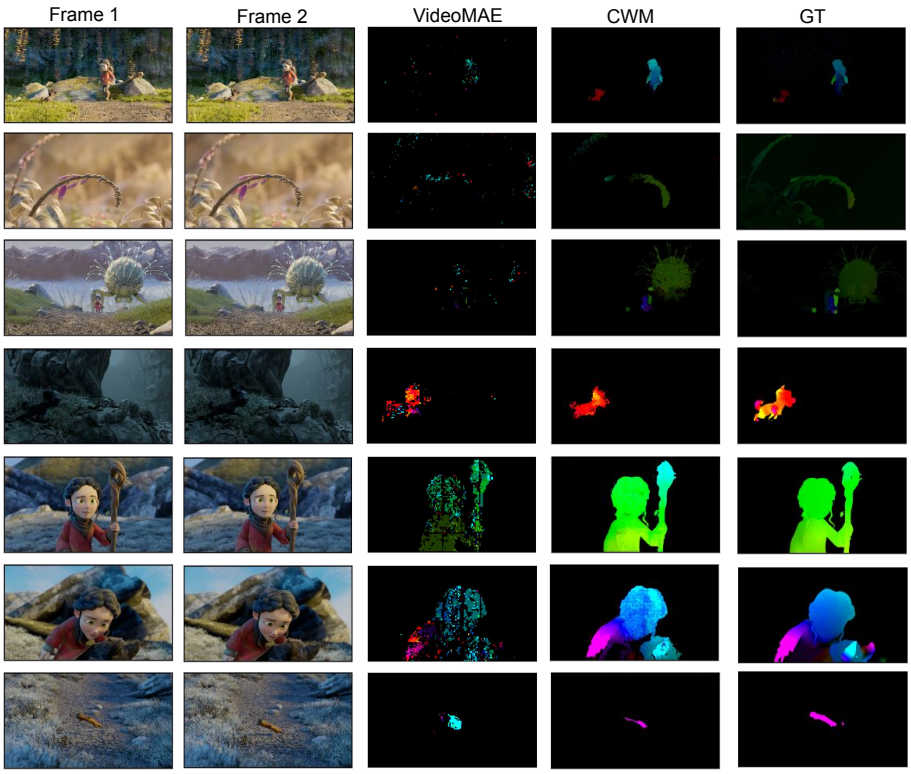


Fig. 9: Additional Optical Flows extracted on the Spring dataset. 23

Fig. 10a: Pseudo-masks extracted in a zero-shot manner on COCO training images. FreeSOLO extracts dense masks and removes redundancy via mask non-maximum-suppression (NMS). CutLER and CWM extracts at most 3 masks per image. The pseudo-masks are used as self-supervision signals for training downstream detectors. 24

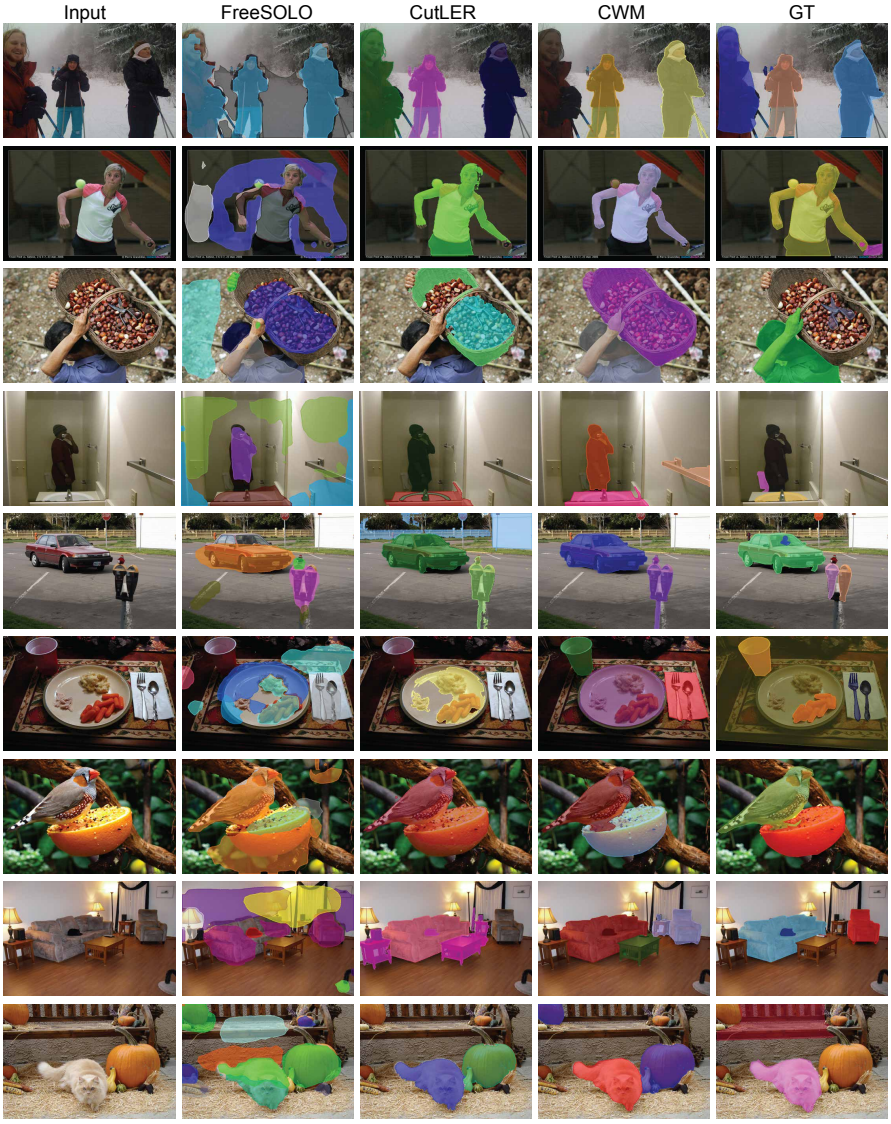
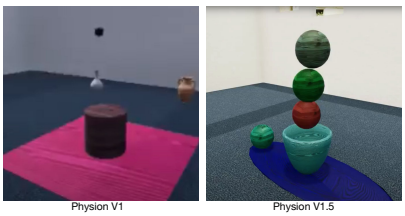


Fig. 10b: More qualitative results on pseudo-masks extracted in a zero-shot manner on COCO training images. 24

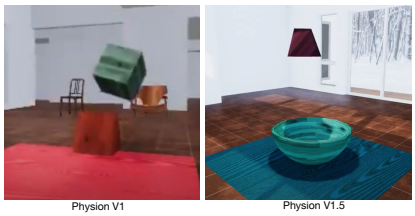
Fig. 11a: Unsupervised segmentation results from the distilled instance segmenter. We show the results on COCO validation images. 24

Fig. 11b: More unsupervised segmentation results from the distilled instance segmenter. 24

(a) Diverse lighting conditions



(b) Improved visual realism: high resolution rendering



(c) Improved visual realism: rendering better shadows

(d) New indoor environments

Fig. 12: Physion v1.5 features key improvements over v1 such as a) enhanced diversity of lighting conditions by employing HDRI skyboxes, b) higher resolution rendering, c) improved rendering of shadows and c) an additional indoor environment ("Craft room"). 25

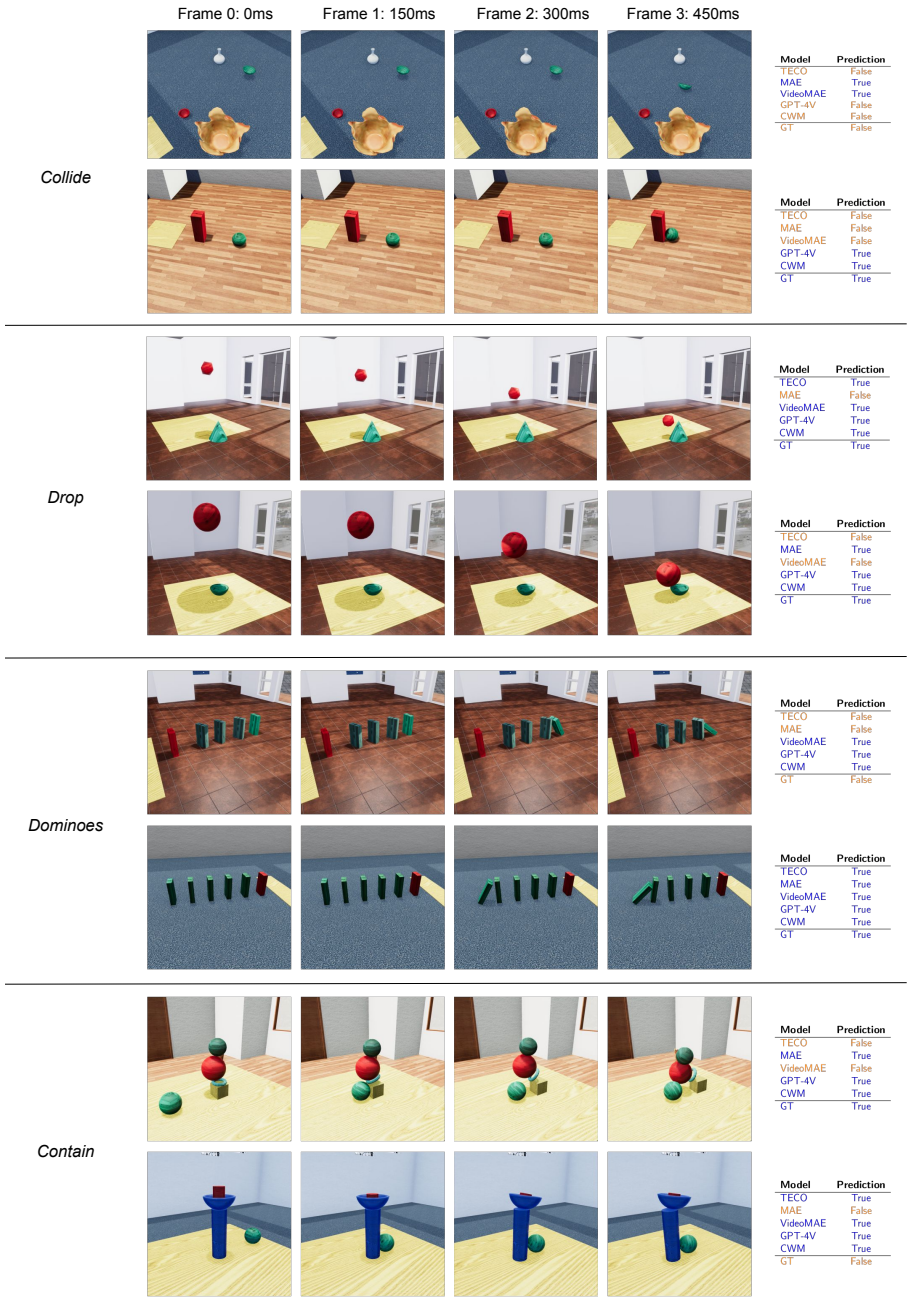


Fig. 13: Model predictions on the OCP task. The input frames sampled at a frame gap of 150ms are shown on the left and the model predictions are shown on the right. We compare against the best performing models in each model category outlined in Table 7. 25

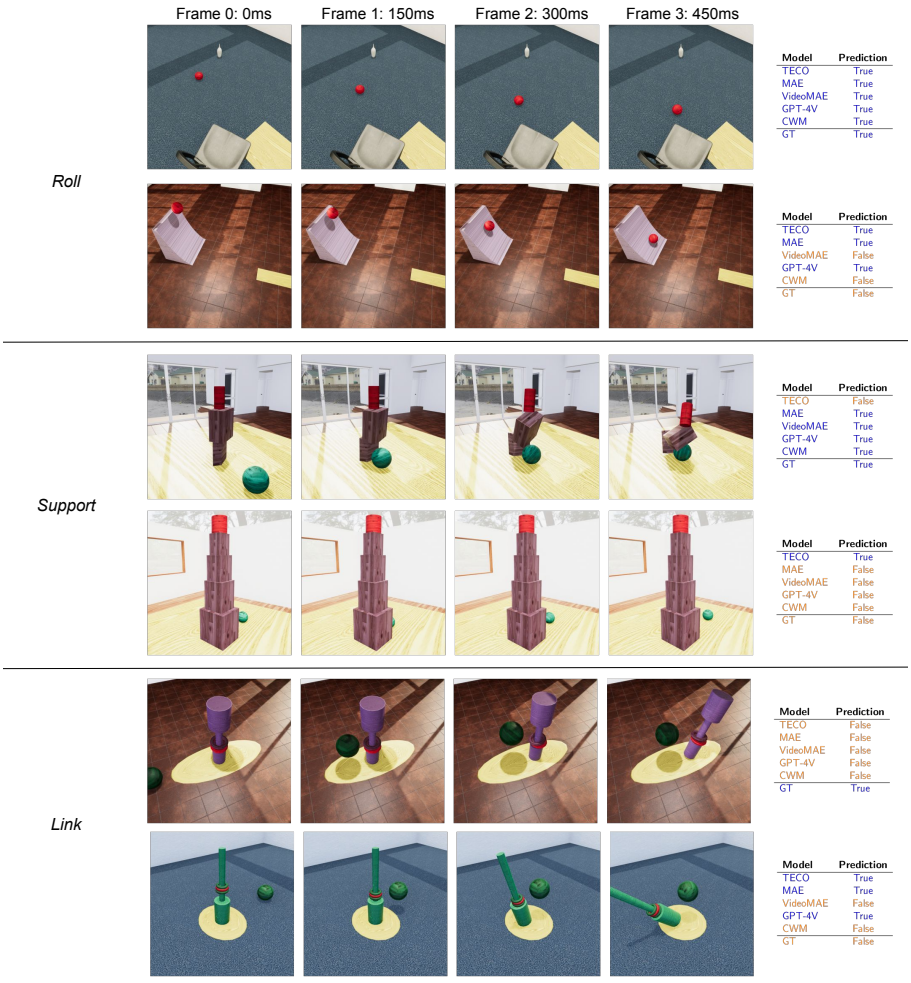


Fig. 14: More model predictions on the OCP task. 25

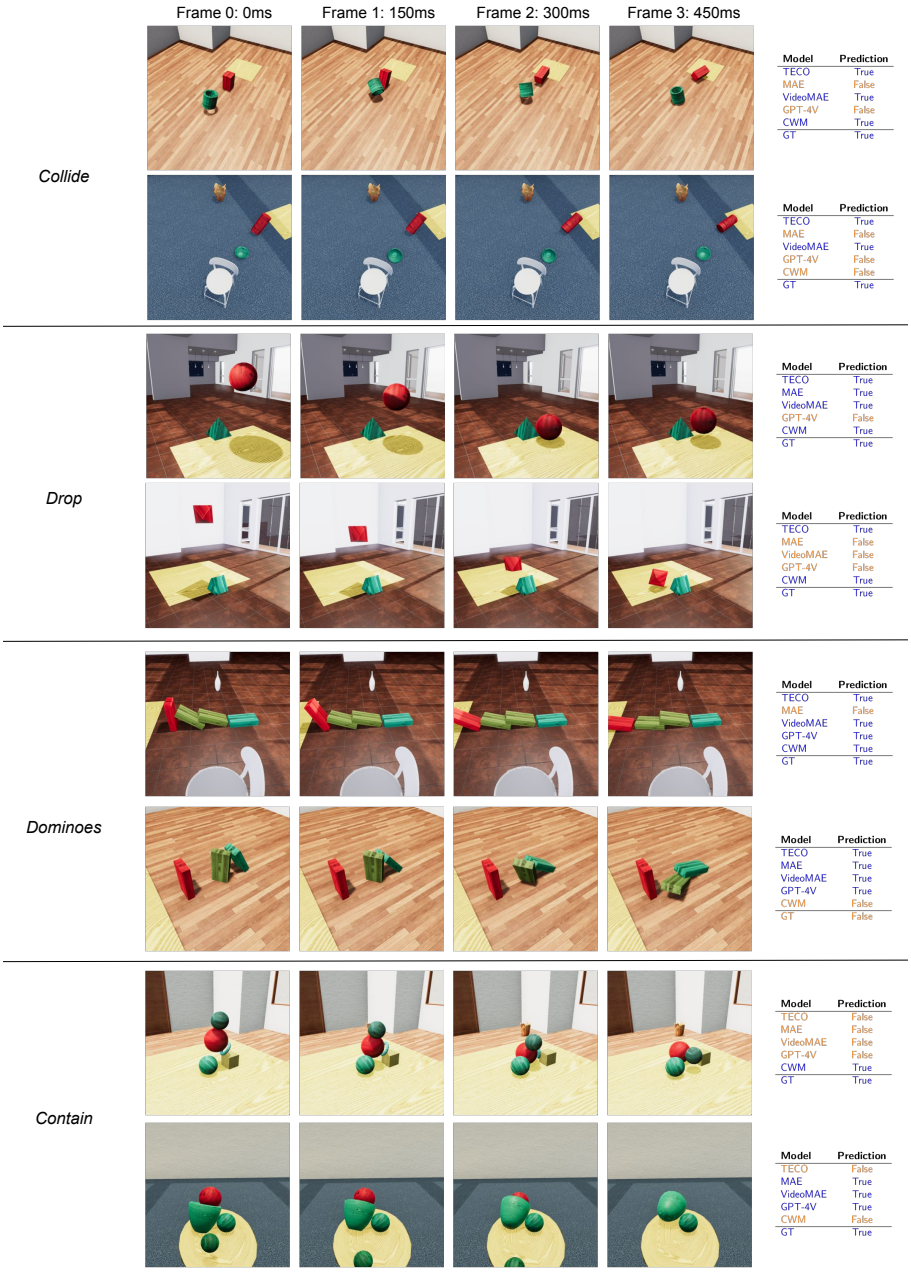


Fig. 15: Model predictions on the OCD task. The input frames sampled at a frame gap of 150ms are shown on the left and the model predictions are shown on the right. We compare against the best performing models in each model category outlined in Table 8. 25

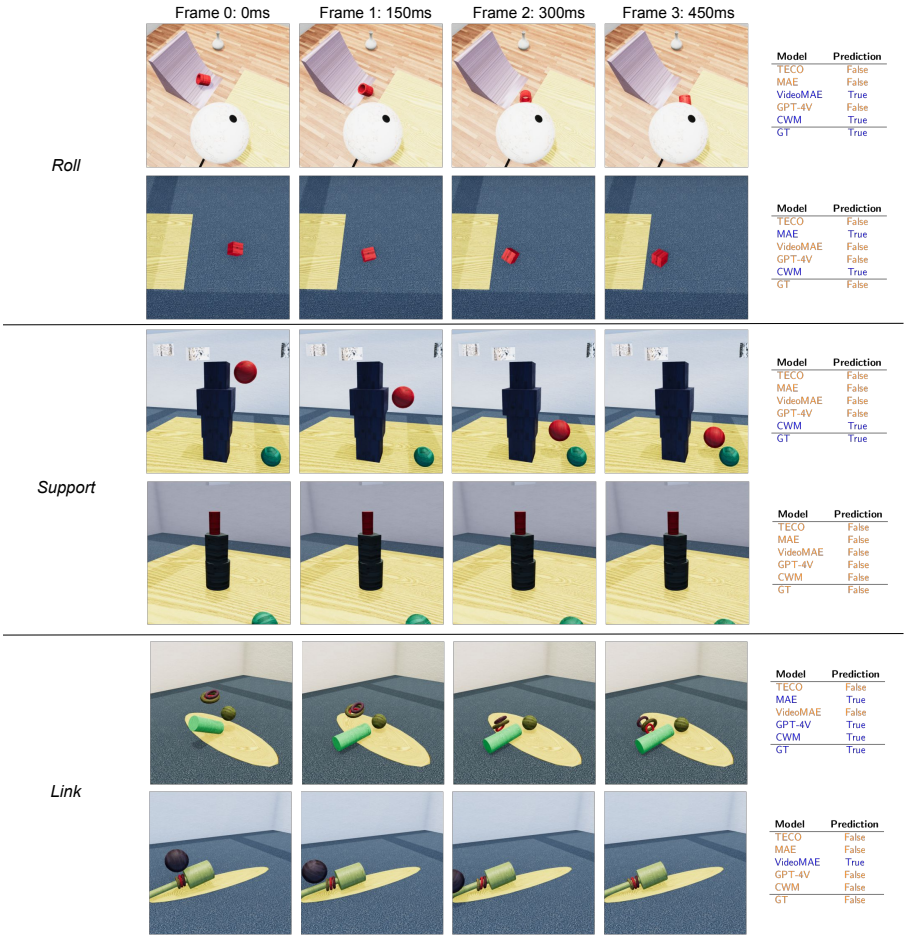


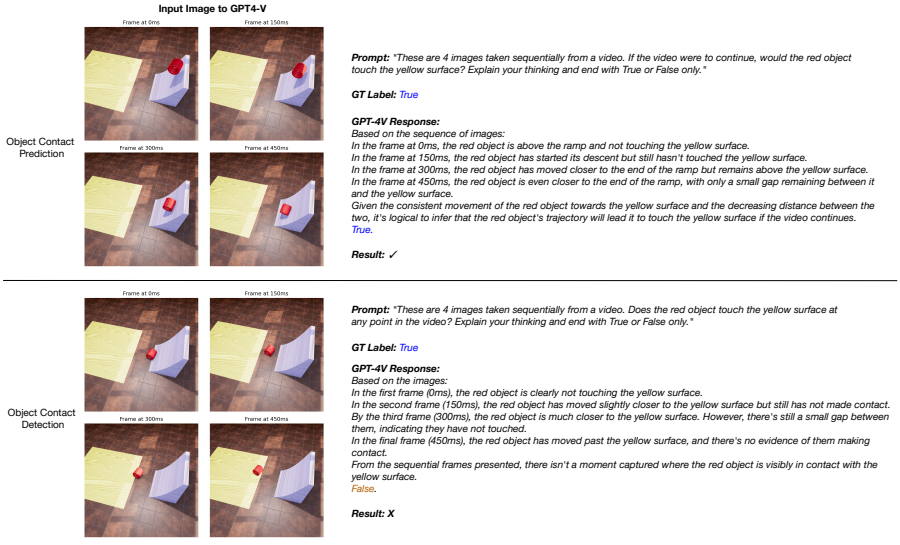
Fig. 16: More model predictions on the OCD task. 25

Table 7: Physion v1.5 scenario-wise OCP results. We compare CWM to four classes of baseline methods across different architectures on the OCP task. In the main text, we presented the scores averaged across all seven scenarios. Here we provide a detailed breakdown of performance for each specific scenario. For a strictly fair comparison we train VideoMAE* with the same patch size and number of frames as CWM. 25

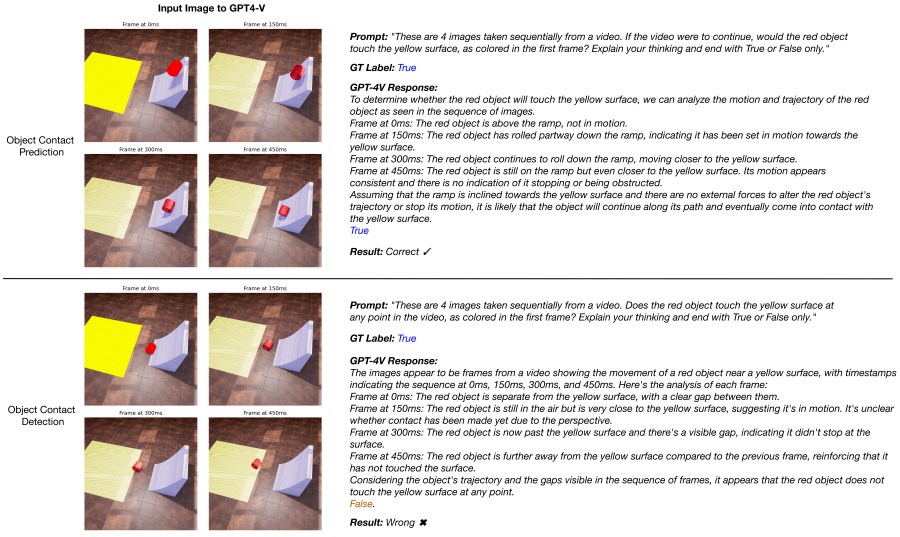
method	arch	OCP by scenario ↑							Avg. OCP ↑
		collide	drop	support	link	roll	contain	dominoes	
video prediction models									
MCVD [74]	UNet	73.3	65.0	70.7	59.2	51.0	66.7	57.9	63.4
R3M [56]	Res50	79.0	61.8	70.1	66.9	**66.2**	62.1	67.3	67.6
FitVid [4]	VAE	65.0	68.2	71.5	59.9	54.1	60.8	70.7	64.3
TECO [82]	vq-gan	75.9	70.6	74.4	65.2	59.1	72.4	67.5	69.3
self-supervised image representation models									
DINO [13]	ViT-B	79.0	72.6	81.0	**72.0**	61.8	69.3	69.2	72.1
DINOv2 [60]	ViT-B	78.1	70.7	80.3	70.7	64.3	70.6	70.4	72.2
DINOv2 [60]	ViT-L	81.0	68.8	82.3	68.8	61.1	69.9	73.6	72.2
DINOv2 [60]	ViT-g	80.0	74.5	81.0	66.2	61.8	74.5	71.1	72.7
MAE [40]	ViT-B	80.0	72.6	78.9	70.1	64.3	68.0	74.2	72.6
MAE [40]	ViT-L	80.0	73.2	81.0	69.4	58.0	69.3	70.4	71.6
MAE [40]	ViT-H	**83.8**	72.0	**84.4**	70.1	61.8	69.3	71.7	73.3
MAE [40]	ViT-B	81.9	70.7	83.0	68.8	59.9	67.3	73.0	72.1
MAE [40]	ViT-L	83.8	70.7	81.0	68.2	59.9	70.6	74.2	72.6
self-supervised video representation models									
VMAE [73]	ViT-B	74.3	74.5	83.0	65.6	61.8	71.2	74.2	72.1
VMAE* [73]	ViT-B	80.0	71.3	82.3	70.1	58.6	72.5	76.7	73.2
VMAE [73]	ViT-L	79.0	73.9	82.3	66.9	65.0	72.5	75.5	73.6
VMAE [73]	ViT-H	81.0	73.2	81.6	70.7	63.1	70.6	74.2	73.5
VMAEv2 [75]	ViT-g	77.1	75.2	83.0	65.0	62.4	70.6	72.3	72.2
V-JEPA [73]	ViT-L	80.1	68.8	84.3	69.4	62.4	73.8	74.2	73.4
vision-language models									
GPT4-V [59]	-	52.7	46.5	58.5	54.8	56.2	56.1	46.5	52.9
CWM	ViT-B	82.9	75.2	83.7	70.7	63.7	**77.8**	77.4	75.9
CWM	ViT-L	**83.8**	74.5	**84.4**	71.3	65.0	75.8	**78.0**	**76.1**

Table 8: Physion v1.5 scenario-wise OCD results. We compare CWM to four classes of baseline methods across different architectures on the OCD task. In the main text, we presented the scores averaged across all seven scenarios. Here we provide a detailed breakdown of performance for each specific scenario. For a strictly fair comparison we train VideoMAE* with the same patch size and number of frames as CWM. 25

method	arch	OCD by scenario ↑							Avg. OCD ↑
		collide	drop	support	link	roll	contain	dominoes	
video prediction models									
MCVD [74]	UNet	82.9	74.5	95.9	75.8	68.8	77.8	89.9	80.8
R3M [56]	Res50	83.8	72.0	90.5	72.0	72.0	70.6	86.2	78.1
FitVid [4]	VAE	58.9	56.7	63.1	63.2	60.2	55.5	58.8	59.5
TECO [82]	vq-gan	87.0	77.5	87.5	70.7	72.6	76.3	95.0	80.9
self-supervised image representation models									
DINO [13]	ViT-B	87.6	79.6	95.2	81.5	76.4	80.4	**96.9**	85.4
DINOv2 [60]	ViT-B	89.5	**84.7**	**96.6**	**86.6**	76.4	79.1	**96.9**	87.1
DINOv2 [60]	ViT-L	91.4	79.0	**96.6**	84.1	73.9	77.8	95.6	85.5
DINOv2 [60]	ViT-g	91.4	80.3	95.2	83.4	70.1	77.1	94.3	84.6
MAE [40]	ViT-B	86.7	76.4	92.5	77.7	71.3	72.5	93.7	81.6
MAE [40]	ViT-L	86.7	75.8	93.9	80.3	70.7	73.2	95.6	82.3
MAE [40]	ViT-H	84.8	75.2	92.5	76.4	67.5	73.2	96.2	80.8
MAE [40]	ViT-B	86.7	75.8	91.2	76.4	70.7	74.5	**96.9**	81.7
MAE [40]	ViT-L	89.5	76.4	93.2	77.1	69.4	72.5	95.0	81.9
self-supervised video representation models									
VMAE [73]	ViT-B	91.4	78.3	94.6	80.3	76.4	83.0	95.6	85.7
VMAE* [73]	ViT-B	93.3	79.6	94.6	82.2	75.2	82.4	96.2	86.2
VMAE [73]	ViT-L	95.2	78.3	95.2	82.8	75.2	80.4	95.6	86.1
VMAE [73]	ViT-H	95.2	79.6	95.2	84.7	79.0	81.7	**96.9**	87.5
VMAEv2 [75]	ViT-g	91.4	78.3	91.8	84.7	73.9	81.7	93.1	85.0
V-JEPA [73]	ViT-L	93.3	84.7	95.9	83.4	73.2	83.0	95.6	87.0
vision-language models									
GPT4-V [59]	-	52.9	53.0	54.2	60.7	56.2	56.1	49.7	54.7
CWM	ViT-B	96.2	82.2	95.9	85.4	81.5	**86.3**	96.2	**89.1**
CWM	ViT-L	**96.2**	83.4	**96.6**	84.1	81.5	83.0	96.2	88.7

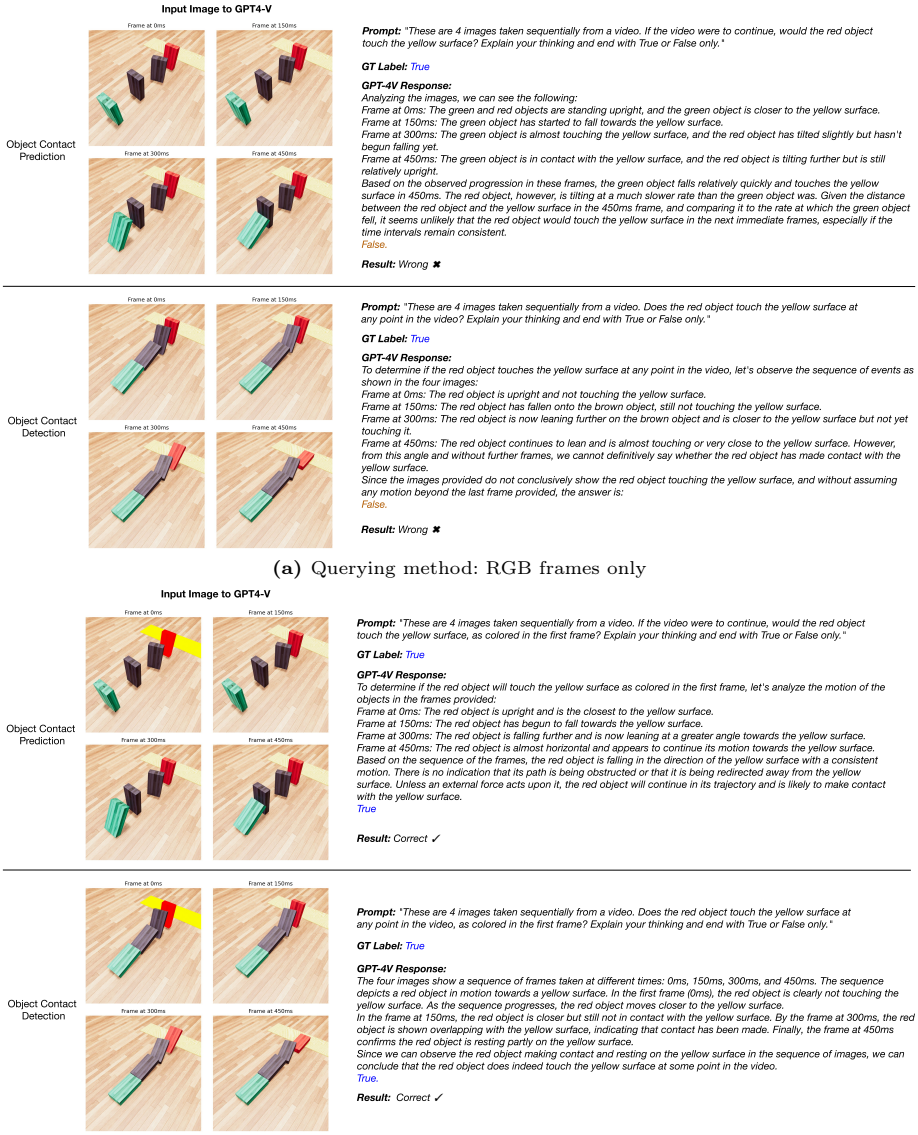


(a) Querying method: RGB frames only



(b) Querying method: RGB frames with objects of interest highlighted

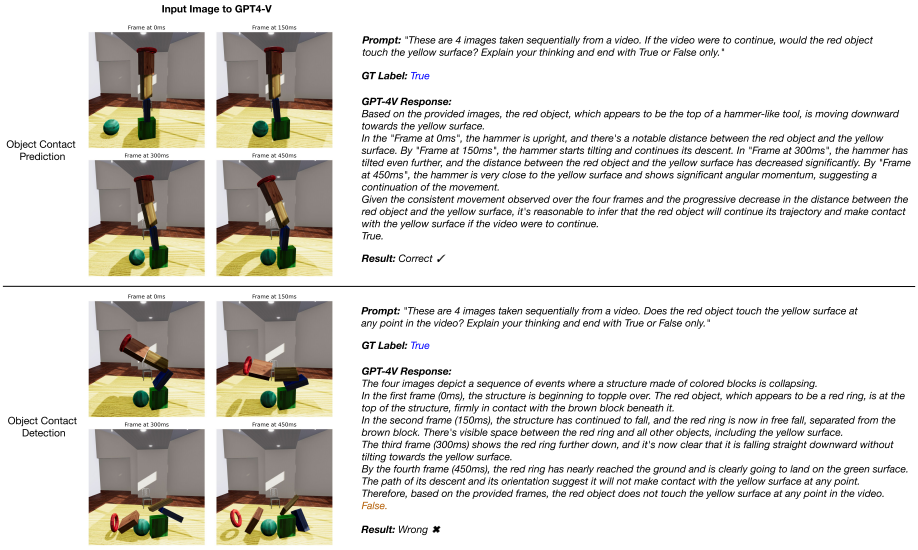
Fig. 17: GPT-4V results on the OCP and OCD tasks in Physion v1.5. The input image prompt is shown on the left and the text prompt, GPT-4V response and results are shown on the right. We show the results of two different querying strategies: a) with RGB frames only and b) RGB frames with ground truth segment map overlayed on the objects of interest. 26



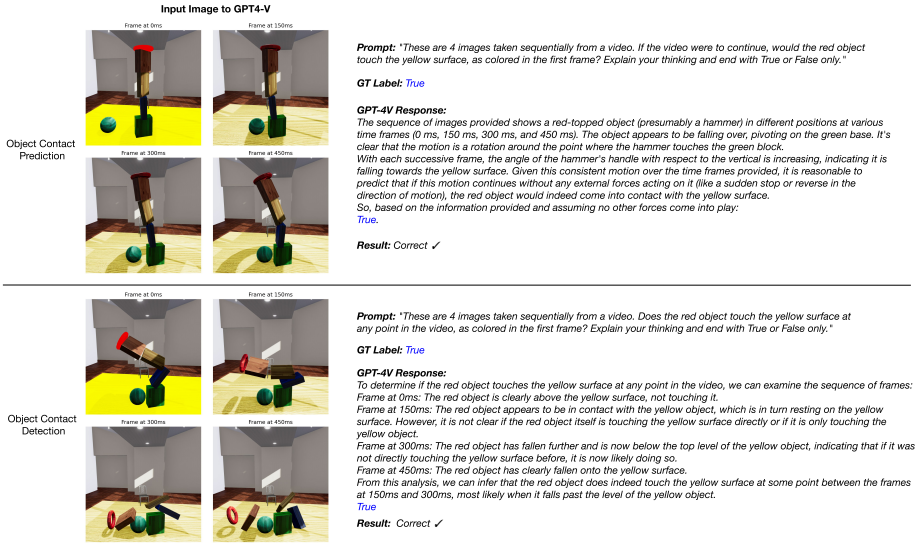
(a) Querying method: RGB frames only

(b) Querying method: RGB frames with objects of interest highlighted

Fig. 18: More GPT-4V results. 26

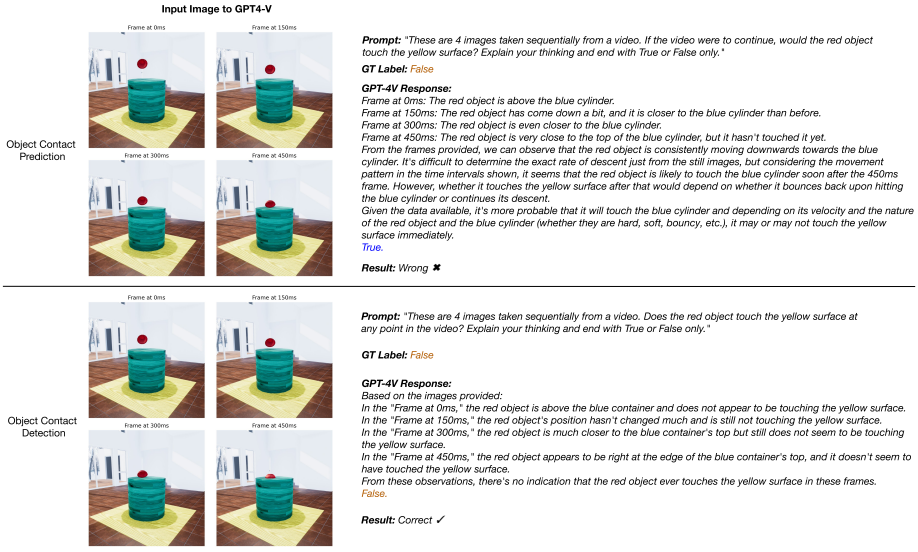


(a) Querying method: RGB frames only

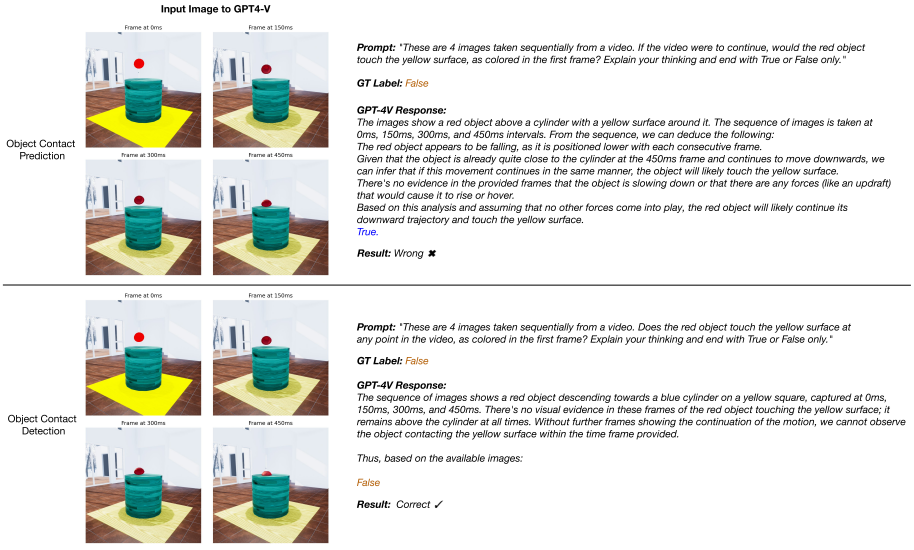


(b) Querying method: RGB frames with objects of interest highlighted

Fig. 19: More GPT-4V results. 26



(a) Querying method: RGB frames only



(b) Querying method: RGB frames with objects of interest highlighted

Fig. 20: More GPT-4V results. 26