COMPOSITIONAL 4D DYNAMIC SCENES UNDERSTANDING WITH PHYSICS PRIORS FOR VIDEO QUESTION ANSWERING

Anonymous authors
Paper under double-blind review

ABSTRACT

For vision-language models (VLMs), understanding the dynamic properties of objects and their interactions in 3D scenes from videos is crucial for effective reasoning about high-level temporal and action semantics. Although humans are adept at understanding these properties by constructing 3D and temporal (4D) representations of the world, current video understanding models struggle to extract these dynamic semantics, arguably because these models use cross-frame reasoning without underlying knowledge of the 3D /4D scenes. In this work, we introduce **DynSuperCLEVR**, the first video question answering dataset that focuses on language understanding of the dynamics properties of 3D objects. We concentrate on three physical concepts — *velocity*, *acceleration*, and *collisions* within 4D scenes. We further generate three types of questions, including factual queries, future prediction, and counterfactual reasoning that involve different aspects of reasoning on these 4D dynamics properties. To further demonstrate the importance of explicit scene representations to answer these 4D dynamics questions, we propose **NS-4DPhysics**, a **N**eural-**S**ymbolic VideoQA model integrating **Physics** prior for **4D** dynamics properties with explicit scene representation of videos. Instead of answering the questions directly from the video text input, our method first estimates the 4D world states with a 3D generative model powered by physical prior, and then use the neural symbolic reasoning to answer the questions on top of the 4D world states. Our evaluation on all three types of questions in DynSuperCLEVR show that previous video question answering models and large multimodal models struggle with questions about 4D dynamics, while our NS-4DPhysics outperforms previous state-of-the-art models significantly.

1 INTRODUCTION

Visual question answering (VQA) is a comprehensive task to assess how well machine learning models can identify objects, understand their relationships, and perform reasoning with multimodal input. When it comes to video question answering (VideoQA), models must not only capture the static features in frames but also understand temporal dynamics, such as object movements and interactions over time, especially in 3D space. Despite the recent progress in multimodal foundation models, understanding these dynamic features remains challenging (Ma et al., 2024).

A series of studies in cognitive science (Hamrick et al., 2016; Ullman et al., 2018) has found that humans excel at understanding the dynamics and interactions in the physical world. This enables humans to understand the spatial and temporal relationships between objects, predict future object states and interactions, and ultimately perform complex tasks such as planning and manipulation in the 3D world. For video question answering, it is also important to incorporate these dynamic features as an integral part of video understanding.

However, existing video question answering datasets have primarily focused on high-level temporal semantics, such as human activities (Fabian Caba Heilbron & Niebles, 2015; Goyal et al., 2017), due to a lack of 3D/4D annotations. Such evaluations are often susceptible to biases and shortcuts in natural videos, so models may perform quite well without truly understanding key dynamic properties (Ma et al., 2024). Another line of work exploits synthetic environments and studies

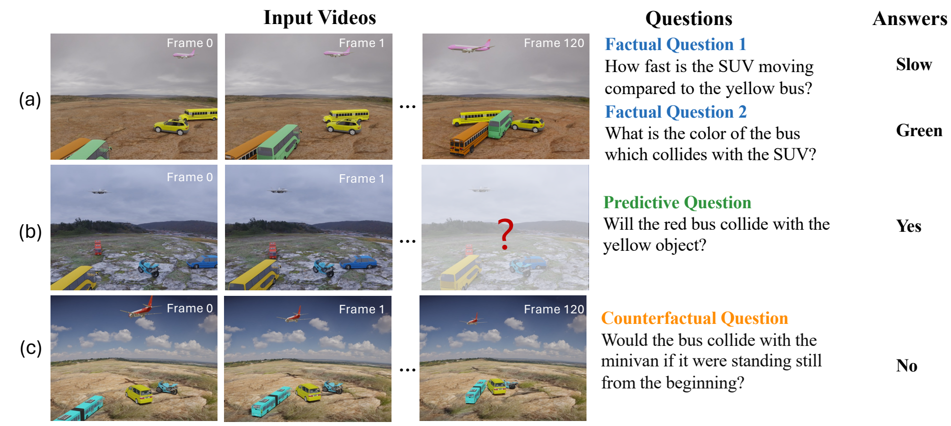


Figure 1: We propose DynSuperCLEVR to study the 4D dynamics properties of objects and their collisions. We also design three types of questions. Factual questions and counterfactual will take the whole 120 frames as input, and the predictive questions will take the first 30 frames (see Sec. 3.4)

how models understand dynamic and physical properties, given 3D ground truth obtained from simulators (Yi et al., 2019; Chen et al., 2022; Bear et al., 2021; Tung et al., 2023; Ates et al., 2022; Patel et al., 2022; Girdhar & Ramanan, 2020; Zheng et al., 2024). However, these datasets often consider only simplified settings, either lacking diverse physical properties in simulations (Ates et al., 2022; Yi et al., 2019) or lacking language integration (Bear et al., 2021; Tung et al., 2023). Moreover, these datasets employed rudimentary rendering with clear backgrounds and toy-like 3D assets, creating a significant domain gap between synthetic videos and real-world natural videos. While many conceptual findings have been derived from experiments on these datasets, these results lack practicality for understanding complex physical events in the real world. Without realistic rendering and natural language support, these datasets are also not suitable for analyzing the 4D dynamics understanding of recent large multimodal models.

To quantitatively study 4D dynamics understanding and to analyze the limitations of current video understanding models, we introduce **DynSuperCLEVR**, the first video question answering (VideoQA) dataset that focuses on the language understanding of the dynamic 4D properties of objects. Specifically, our dataset studies three crucial dynamic properties of 3D objects – velocities, accelerations, and collisions, and designs natural language questions for three types of reasoning, *i.e.*, factual, future prediction, and counterfactual. Notably, by introducing acceleration, we achieve a more complex physical dynamics scene compared to the previous data generator (Greff et al., 2022). Lastly, to address the synthetic-to-real domain gap in previous video benchmarks (Yi et al., 2019; Tung et al., 2023) and to enable zero-shot evaluation of large multimodal models on our dataset, we improve the realism of our videos by introducing diverse appearances of objects and realistic backgrounds with new materials and textures.

To demonstrate the importance of an explicit scene representation in answering these 4D dynamics questions, we present **NS-4DPhysics**, a neural-symbolic model that reasons about these dynamics by first estimating explicit 4D scene representations. Our NS-4DPhysics consists of two key modules, a 4D scene parsing module followed by a symbolic reasoning module. The 4D scene parsing module learns a robust 3D generative model (Ma et al., 2022; Wang et al., 2023b) of the scene and captures the dynamics of objects with a physical prior. With a 3D generative representation of the scene, our scene parsing module is robust to partial occlusion and general well to unseen 3D poses and object appearances during training, which are very common in natural videos.

We test a wide range of state-of-the-art VideoQA models on our DynSuperCLEVR dataset, including neural symbolic model (Yi et al., 2019; Wang et al., 2024), standard end-to-end models (Perez et al., 2018), large multimodal models (Xu et al., 2024; Lin et al., 2023), and proprietary model GPT-4o. Results show that these models struggle to answer challenging questions about dynamic properties about the 4D scenes, due to a lack of explicit knowledge of the 3D world or 3D dynamics of objects. Meanwhile, our NS-4DPhysics model outperforms previous state-of-the-art models by a wide margin, across different dynamic properties and types of reasoning questions. This demonstrates the importance of developing an explicit 4D dynamics representation for multimodal agents to understand physical events presented in the videos and to foresee upcoming events. Our contributions are as follows: (1) We present DynSuperCLEVR, the first VideoQA dataset that focuses on language understanding of the dynamics properties of objects (velocity, acceleration) and multi-object interactions (collisions). (2) We propose NS-4DPhysics, a neural-symbolic model that first reconstructs the 4D scene with a dynamic 3D generative model with physical priors as the perception module, and then reasons about dynamics over the explicit 4D scene representation. (3) Extensive results on different settings of DynSuperCLEVR reveal key pitfalls of state-of-the-art VideoQA models, including those powered by LMMs and industrial training data. We further demonstrate that NS-4DPhysics outperforms other VideoQA baselines by a wide margin, showing the advantages of learning explicit 4D dynamic representations.

2 Related Work

Video question answering. Video questions answering (VideoQA) is a challenging task because models must not only detect and identify objects from static images, but also track and infer objects' changes and interactions over a sequence of frames. A number of VideoQA datasets annotate question-answer pairs on natural videos (Xue et al., 2017; Yang et al., 2022; Majumdar et al., 2024; Wu et al., 2024). However these datasets are not suitable for studying dynamic physical properties as natural videos contain limited object interactions. Another line of works focused on physical reasoning in simulated environments (Yi et al., 2019; Chen et al., 2022; Ding et al., 2021; Tung et al., 2023). However, these datasets are either built in restricted settings (Yi et al., 2019; Chen et al., 2022) or lack of real dynamics and forces common in real world (Tung et al., 2023). In order to further challenge and explore the limitations of current video-text models on dynamics reasoning, we develop DynSuperCLEVR with improved realism for both video quality and physics simulation.

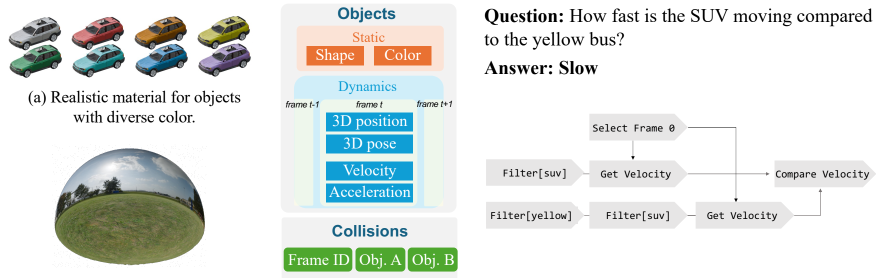
Video-text models. With the availability web-scale image-text or video-text paired datasets (Schuhmann et al., 2022; Bain et al., 2021), recent video-text models (Wang et al., 2022; Lin et al., 2023; Xu et al., 2024) adopted heavy multi-modal pretraining and achieved improved results on a wide range video understanding tasks. However, these models often exploit biases and shortcuts for dynamic reasoning, failing to capture the physical intrinsics that drive object movements and interactions. Our NS-4DPhysics incorporated a 4D dynamic scene representation in our scene parser, allowing compositional reasoning of various physical events in the video sequence.

Physical scene understanding. Understanding the physical events in a dynamic 4D scene or inferring the future state is hard as it requires a model to apply physical principles to explain the observed events or simulate the future outcomes. Previous studies explored physics engines for simulation and learning (Kadambi et al., 2023; Battaglia et al., 2013), or integrating a differentiable physics engine in deep learning models (Wu et al., 2015; 2017; Kyriazis & Argyros, 2013). Another line of works learned physical properties from a compositional scene (Hamrick et al., 2016; Ullman et al., 2018; Ding et al., 2021; Chen et al., 2022; Zheng et al., 2024). However, previous methods were limited to simple scenes where physics models provide good approximations of the world. We extend our scope to more realistic scenes with real-world objects and complex simulations, and enable physical scene reasoning with our dynamic 3D compositional reasoning.

3D generative models. Our model is built on top of the previous 3D generative models for image classification (Jesslen et al., 2023), 6D pose estimation (Ma et al., 2022), and 3D-aware visual questions answering (Wang et al., 2023a). In image understanding, these generative models learn a compositional representation of 3D objects with feature activations on vertex, conducting the scene understanding tasks with analysis-by-synthesis. As found in Wang et al. (2023a), the advantage of these methods is the robustness in recognizing the 3D objects from scenes, compared with the discriminate baseline such as FasterRCNN (Ren et al., 2016). We extended the previous 3D compositional models to video understanding task, which learns a 4D dynamic scene representation from video. Such representation allows us to reconstruct world states at each timestep, analyze the trajectories of objects and reason about the physical events.

3 Dataset

To investigate 4D dynamic properties in VideoQA tasks, we introduce the **DynSuperCLEVR** dataset. This synthetic dataset provides fully annotated 4D scene structures for videos, along with the reasoning steps to answer corresponding questions. We concentrate on the 4D dynamics properties of objects — including `velocity`, `acceleration` in 3D scenes and their comparison. We also introduce the `collision` event in 3D space, which is more challenging than on the 2D ground plane. The collision shows how the dynamic properties will affect the interactions among objects.



(a) Realistic material for objects with diverse color.

(b) Environment map in dome shape as background.

(c) Structure of scene annotations for objects and events.

(d) Example of program execution for questions answering on dynamical properties

Figure 2: An illustration of the construction of DynSuperCLEVR. (a) We use the same 3D object meshes from SuperCLEVR but generate more realistic textures for different colors. (b) The background is created by mapping a real image environment map onto a dome shape. (c) Our video data is fully annotated with 4D dynamic scene structure, containing static and dynamic properties for objects and collision components. (d) For each question, we design new operation programs for 4D dynamics properties, which can be executed on the scene structure to answer the questions.

As an advanced video reasoning task, we design questions across three levels — (1) **factual questions**, which query properties and events directly from the visible frames, (2) **predictive questions**, which require forecasting future collisions based on the current 4D dynamics and scenes, and (3) **counterfactual questions**, which challenge models to imagine how changes in the 4D dynamics would alter the outcomes.

In this section, we first introduce the three key components for designing the video data in our proposed dataset: (1) the design of more realistic texture for objects and backgrounds (Sec.3.1), (2) the 4D dynamical scene annotations involving 4D dynamics properties for objects (Sec.3.2) and their collisions events (3) the rendering of the video and physical simulations process (Sec.3.3). Then in Sec.3.4, we describe our three types of questions at different reasoning level and the reasoning program for question answering.

3.1 3D Scene construction

3D Objects. We use the same 3D objects from SuperCLEVR (Li et al., 2023), which include five vehicle categories (car, plane, bicycle, motorbike, bus) with 21 sub-types, but we regenerate the texture for each mesh model to improve the realism. For each color label (gray, red, brown, yellow, green, cyan, blue, and purple), we create 6 variants and change the faces color of the object meshes. Based on the part annotation in SuperCLEVR, only the primary body receives color edits, while others are updated with natural elements like black wheels, transparent windows, or red tail lights on cars (see Fig. 2a). More examples of all textured 3D objects are provided in the Appendix A.

Backgrounds. Compared with the previous synthetic dataset (Yi et al., 2019; Li et al., 2023), we change the background into highly realistic images and lights. As the example in Fig. 2 (b), we use the real HDRI images[1], containing real captured images with environment maps in 509 different scenes. We map the image as material into a dome shape as background to create a 3D view.

3.2 4D Dynamics Scene Annotations

After defining the shape and color of objects as static attributes, we assign them dynamic properties, allowing them to move and interact within the 3D physical world. Each object is defined by its 3D position, pose, velocity, and acceleration, as described below (see Fig. 2(c)).

3D Position and pose: The 3D position (x, y, z) records the exact coordinates of the object at each timestep, specifying its location within the scene. The pose (α, β, γ) describes the object's orientation and rotation in 3D space, defining how the object is aligned relative to its surroundings.

Velocity. The `velocity` (v_x, v_y, v_z) represents the change of the object's 3D position over time, capturing both the speed and direction of motion. As one of the properties included in the question answering, we define three velocity states: `static` ($0m/s$), `slow` ($\leq 3m/s$) and `fast` ($\geq 3m/s$)

[1] `https://polyhaven.com/hdris`

for language description. Given that the exact velocity is often difficult to determine from video alone, we ensure that each scene includes at least one object moving within each velocity state to provide clear distinctions.

Acceleration. The `acceleration` (a_x, a_y, a_z) represents the rate of change of velocity over time in 3D space. Similar with the label for velocity, we define two types of motion for acceleration. Along the x-axis, objects can either accelerate due to an internal engine force or decelerate due to friction. In z direction, the objects can either maintain a constant height (zero acceleration) or downward acceleration with gravity. In our dataset, we simply it in to a binary label: `accelerating` or not, and `floating` or not.

Collision Events. Due to perceptual illusions, reasoning about collisions in 3D space is more challenging than in 2D plane. Each collision involves two `objects`, and we record the `frame ID` when the collision occurs. The collision between objects is a direct result of their 4D dynamic properties. In our dataset, we capture not only the collision events in a given frame but also future collisions from simulation, and counterfactual scenarios, where changes in the 4D dynamic properties at the beginning lead to a new sequence of collision events. This forms the foundation for the three question types, which will be introduced in Sec. 3.4.

3.3 Physical Simulation and Video Generation

The video data is rendered using a combination of the physics engine PyBullet and the Blender renderer. Building on the generation pipeline of Kubric (Greff et al., 2022), we integrate support for the new acceleration feature. At each timestep, we capture the 3D position, pose, velocity, and acceleration of all objects, along with all collisions between objects, as described above.

Each scene has a corresponding counterfactual version. We re-simulate the scene, randomly selecting one object's 4D dynamic property (velocity or acceleration) to modify its initial value. We then record the new property values and the resulting sequence of collision events caused by the changes.

3.4 Questions generation for 4D dynamics properties

Following previous video question answering datasets (Yi et al., 2019; Chen et al., 2022), we develop question templates and generate three types of questions for factual, predictive, and counterfactual reasoning in our DynSuperCLEVR (see Fig. 1). Each question is paired with a corresponding program execution for reasoning (see Fig. 2(d)). Below, we describe the design of the three question types, while the full list of programs and more examples of program execution can be found in the Appendix C.

Factual Questions. Factual questions focus on the direct understanding of static and dynamic properties of objects, their comparisons, and collision events within 3D space in given frames. For our newly proposed 4D dynamic properties, we define questions that query the velocity and acceleration states of objects, or compare their velocities in 3D space at specific moments. These moments are either defined by the start of a frame or the point when a collision event occurs. The questions about collisions are predicting whether two objects have collided or identifying which objects are involved in a collision. We introduce new programs $query\_velocity$, $compare\_velocity$ and $query\_moving\_direction$ for velocity; $query\_accelerating$, $query\_floating$ for acceleration.

Predictive Questions Predictive questions require forecasting future collision events that are not present in the video. In our DynSuperCLEVR, it is crucial for the model to first understand the positions, velocities, and accelerations of objects in 3D space from the video, and then apply reasoning to predict their future trajectories.

Counterfactual Questions. Counterfactual questions challenge the model to reason about hypothetical scenarios by changing the initial 4D dynamic properties of objects. This requires the model to have a strong understanding of how changes in dynamic features (velocity or acceleration) can affect the world states and cause collisions, and apply these to re-simulate the potential outcomes contradicted to the video input.

3.5 Dataset Statistics

We generate 1,000 video clips for training, 100 for validation, and 100 for testing. Each video is 2,000 ms long with a frame rate of 60, resulting in 120 frames per scene. In total, we create 7,850 factual questions, 2,750 predictive questions, and 989 counterfactual questions. Factual questions are

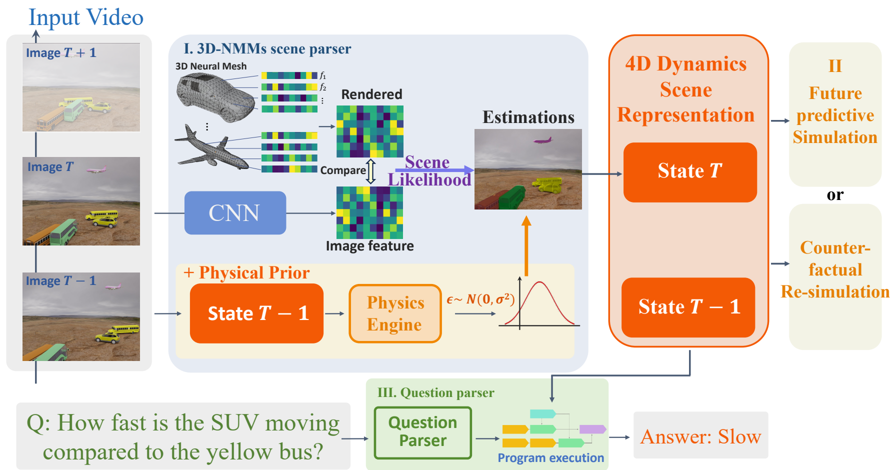


Figure 3: Our NS-4DPhysics has three main components. I: A **3D neural mesh scene parser**, which combined the rendering likelihood with a physics prior, to parses the video into 4D dynamic scene representation. II: The future states or counterfactual states can be simulated with the reconstruction result by the physics engine. III: A **question parser** that processes questions into reasoning programs and then executes the program over the predicted scene representation to answer the questions.

open-ended and can be answered with a few words, while predictive and counterfactual questions are structured as true or false, requiring the model to determine if events will occur.

4 Model

In this section, we introduce NS-4DPhysics, a neural symbolic VideoQA model for solving questions about the object dynamics in 3D space. As illustrated in Fig. 3, the model combines a **3D-aware scene parser** with a **probabilistic physical prior module** to convert video sequences into an explicit 4D scene representation. This explicit representation enables interpretable reasoning, allowing the model to predict future or counterfactual world states through physics simulation. For question answering, we employ a **question parser** to first convert the questions into executable programs, which are then used to answer the questions step by step based on the predicted 4D scene representation.

In the following sections, we introduce the 4D symbolic scene representation in Section 4.1, the scene parser with the physical prior in Section 4.2, future and counterfactual simulation in Section 4.3, and the language model and reasoning program in Section 4.4.

4.1 4D Symbolic Scene Representation

The 4D dynamics scene representation is the world states to reconstruct the video scene annotation shown in Fig. 2c. For each object O, we define its static attributes as **shape** s^i and **color** c^i. The dynamic properties at each timestep t consist of **3D position** T_t, **rotation** (pose) R_t, **velocity** v_t, and **acceleration** a_t, where each is a 3D vector. For an input video, each image frame at timestep t is denoted as $\mathcal{I}_t$. The 4D symbolic scene representation $\mathcal{S}$ is defined as a sequence of object states $\mathcal{S}_t = \{O_t^1, O_t^2, \ldots, O_t^N\}$ over time, where N is the number of objects in the scene.

4.2 Dynamic Scene Parser with Physics Prior

The scene parser takes the videos as input and estimate the 4D dynamic world state $\mathcal{S}_t$ at time step t. This involves predicting the 6D poses of objects in frames, predicting their temporal properties (*e.g.*, velocity and acceleration), and reason about collision events. In this work, we adopt a 3D generative model that learns a feature representation of the scene and predicts 3D world states from an analysis-by-synthesis perspective. We extend Ma et al. (2022); Jesslen et al. (2023) to natural videos by employing a sequence decoding and add a physical prior in our 4D scene representation for the model to understand real-world physical events.

3D-NMMs scene parser 3D neural mesh models (3D-NMMs) (Ma et al., 2022; Jesslen et al., 2023) learns a generative model to produce the 3D feature representation of the objects and parses the scene with analysis-by-synthesis. For each object shape we trains a object mesh $M_s = \{v_i \in \mathbb{R}^3\}_{i=1}^N$ with neural texture $T_s = \{f_i \in \mathbb{R}^c\}_{i=1}^N$, where s is the object category (shape), N is the number of vertices, and c is the feature dimension. A 3D-NMMs can generates the 3D aware scene representation by rendering the neural mesh model $O_s = (M_s, T_s)$ in the given 6d pose α : $F_s(\alpha) = \mathfrak{R}(O_s, \alpha) \in \mathbb{R}^{H\times W\times c}$, with soft rasterization (Liu et al., 2019). By comparing the rasterization of 3D features and the 2D image feature, we can learns the neural textures with ground truth category s and 6D poses α, or estimate α and s during inference. We formulate this render-and-compare process as an optimization of the likelihood model:

$$p(F \mid O_s, \alpha_s, B) = \prod_{i\in\mathcal{FG}} p(f_i \mid O_s, \alpha_s) \prod_{i\in\mathcal{BG}} p(f_i' \mid B) \tag{1}$$

where $\mathcal{FG}$ and $\mathcal{BG}$ are the set of foreground and background locations on the 2D feature map and f_i is the feature vector of F at location i. Here the foreground and background likelihoods are modeled as Gaussian distributions.

In our NS-4DPhysics, α is the 3D position T_t and rotation R_t of each objects for time step t, which is estimate independently for each video frame (i.e. image level). The training phase learns the CNN extractor for feature F, the neural mesh O_s and the background feature B jointly (see Appendix D.1 for details). During inference, by maximizing the likelihood function Eq. 7 on each frame, we can obtain the T_t and R_t for predicted objects.

Physical prior The dynamical 3D world states predicting from the image frame does not consider the consistency in time dimension. As our algorithm are predicting $\mathcal{S}_t$ for each time step sequentially with maximizing likelihood function Eq. 7, one strategy is initializing $\mathcal{S}_t$ with the predicted $\mathcal{S}_{t-1}$ in gradient descent. However, this only fit to the case when the objects are moving uninterruptedly. In the complex dynamical 3D scenes, the trajectory of objects can change dramatically due to their interactions such as collision. These require a comprehensive prior knowledge of the physics world.

It's nontrivial to directly model the physical functions. Many previous studies incorporate physical engines within neural networks only applied to simple shapes like cubes, as the complex shapes of 3D objects and the interaction process are hard to model. On the other hand, computational physical engine(Coumans & Bai, 2016) excel at modeling physical functions for any known shapes but are not differentiable and challenging to integrate into the de-rendering process during inference.

In our method, we modify a discriminative physics engineering PE$(\cdot)$ into a probabilistic model by assuming $(R_t, T_t) = \text{PE}(\hat{R}_{t-1}, \hat{T}_{t-1}) + \varepsilon$ after having $\hat{R}_{t-1}, \hat{T}_{t-1}$. Here we take $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$. Inspired by the idea from Bayes-Kalman filtering (Salzmann & Urtasun, 2011), we integrate a physics prior of (R_t, T_t) into the likelihood function as a correction term after Eq. 7. Formally, the prior $q(R_t, T_t \mid \hat{R}_{t-1}, \hat{T}_{t-1})$ can be computed as:

$$(R_t, T_t \mid \hat{R}_{t-1}, \hat{T}_{t-1}) \sim \mathcal{N}(\text{PE}(\hat{R}_{t-1}, \hat{T}_{t-1}), \sigma^2 I), \tag{2}$$

$$q(R_t, T_t \mid \hat{R}_{t-1}, \hat{T}_{t-1}) = C \exp\left(-\frac{1}{2\sigma^2}\left[(R_t, T_t) - \mu\right]^T \left[(R_t, T_t) - \mu\right]\right), \tag{3}$$

where $\mu = \text{PE}(\hat{R}_{t-1}, \hat{T}_{t-1})$, $C = 1/\sqrt{(2\pi)^k |\sigma^2 I|}$.

Finally, the rotation R_t and translation T_t can be estimated by maximizing the joint likelihood of the rendering likelihood in Eq.7 and the physical prior likelihood in Eq.3:

$$\hat{R}_t, \hat{T}_t = \arg\max_{R_t, T_t} p(F \mid O_y, T_t, R_t, B) \cdot q(R_t, T_t \mid \hat{R}_{t-1}, \hat{T}_{t-1}). \tag{4}$$

Dynamics attributes. After estimating the world states (R_t, T_t), we compute the dynamics attributes `Velocity` and `acceleration` by calculating the differences between consecutive translations and velocities over time. To reduce noise, a moving average filter with a window size of 5 is applied.

Static attributes. `Shape` is determined by selecting the mesh model category with the highest likelihood (Wang et al., 2024). For `color` prediction, we crop the object's region from the RGB image and train an additional CNN classifier for color recognition.

Collisions. Collisions are obtained from the physics engine PE$(\cdot)$ at the time of estimating the physical prior. For each collision, we record the time and objects involved.

4.3 FUTURE AND COUNTERFACTUAL SIMULATION

From the explicit 4D scene representation, we can easily simulate the future or counterfactual states of the objects. For either two cases, we apply the physics engine and assign the predicted translation, rotation, velocity or counterfactual condition to all objects as initial conditions and simulate. We visualize the re-simulation results in Sec. 5.4.

4.4 QUESTIONS PARSING AND PROGRAM EXECUTION

After having the 4D scene representation from the scene parser, we can parse the question into reasoning programs, which is then executed on the scene representation to predict the answer. The question parser follows previous work (Yi et al., 2019; Chen et al., 2022), where an LSTM sequence-to-sequence model is trained to parse the question into its corresponding program.

5 EXPERIMENTS

5.1 EXPERIMENT SETUP

Baseline models. We select the representation models from the following 3 categories as baseline models on DynSuperCLEVR. (1) The simple classification-based methods: CNN+LSTM and FiLM. We extract frame-level features and average them over the time dimension. We encode questions with the last hidden state from an LSTM, and then concatenate the two features to predict answers. (2) Neural symbolic models: NS-DR and PO3D-VQA, which represent the model with explicit 2D / 3D scene representation for question answering, compared with our 4D representation; (3) Video large language models (Video-LLMs): including a fine-tuned InternVideo, and Video-LLaVA, PLLaVA, GPT-4o for zero-shot evaluation with in-context learning. Please refer to the appendix for detailed implementations.

NS-4DPhysics implementation. The dynamic scene parser and CNN classifier is trained on the images and 120k images are used for training. The dynamic scene parser uses ResNeXT for the feature extractor and is trained for 50 epochs with batch size 4 on 4 GPUs. The attributes classifier uses ResNet50 and is trained with the cropped images by the ground truth bounding box. During inference, we set the variance of the physical prior as 3.

5.2 VIDEO QUESTION ANSWERING RESULTS

We first compare the NS-4DPhysics with baseline models for the VideoQA task of DynSuperCLEVR, the results are shown in Tab. 1. For GPT-4o, we also prompt GPT-4o to first describe the video, then reason step-by-step before answering questions (GPT-4o+reasoning).

Table 1: Performance on the DynSuperCLEVR testing split for each question type: factual, predictive, and counterfactual. Factual questions are further divided into sub-types: **Vel**ocity, **Acc**eleration, and **Col**lision, with "**All**" representing overall accuracy. The average is taken as the overall accuracy across the three question types. † indicates GPT-assisted zero-shot evaluation. The highest performance among all baseline models is indicated by underlined values.

	Average	Factual				Predictive	Counterfactual
		All	Vel.	Acc.	Col.		
CNN+LSTM	48.03	40.63	41.71	56.79	25.37	56.04	47.42
FiLM (Perez et al., 2018)	50.18	44.07	48.58	53.09	26.87	54.94	51.54
NS-DR (Yi et al., 2019)	51.44	51.44	55.63	46.34	46.86	-	-
PO3D-VQA (Wang et al., 2024)	62.93	61.22	62.21	73.17	51.20	65.33	62.24
InternVideo (Wang et al., 2022)	52.62	51.07	59.29	49.08	36.06	54.74	59.18
Video-LLaVA † (Lin et al., 2023)	38.09	37.04	37.62	52.76	23.56	38.78	40.88
PLLaVA † (Xu et al., 2024)	59.24	54.61	55.00	63.80	46.63	67.52	73.47
GPT-4o†	51.59	50.82	51.19	57.67	44.71	54.38	50.00
GPT-4o + reasoning †	56.06	55.50	58.81	57.67	47.12	56.93	58.16
NS-4DPhysics	**82.64**	**87.70**	**88.66**	**83.73**	**88.46**	**85.71**	**74.51**

Comparison with classification-based methods. CNN+LSTM only reaches 40.63%, 56.04% and 47.42% for factual, predictive, and counterfactual questions; and FiLM achieves 44.07%, 54.94% and 51.54%. The performance of these models is notably lower than that of NS-4DPhysics, which achieves 87.70%, 85.71% and 74.51% for factual, respectively. As classification-based methods generally rely on extracting features through CNNs, we demonstrates significant advancements in handling dynamic content with explicit scene representation.

Comparison with the Neural-Symbolic methods The comparison between the Neural-Symbolic models shows significant performance improvement as the dimension of scene representation increases. PO3D-VQA constructs explicit 3D scene representation, leading to a better estimation to the objects' position and poses, which is beneficial to infer the dynamics and interaction compared with the 2D presentation in VR-DR (9.78%) for the factual questions. As pushing the dimension into 4D and considering the physical prior knowledge, our NS-4DPhysics achieves 82.64% overall accuracy which is 19.71% higher than PO3D-VQA.

Comparison with the Video-LLM Although Video-LLMs demonstrate strong video understanding and generalization abilities, they underperform on DynSuperCLEVR. In zero-shot settings, Video-LLaVA achieves an overall accuracy of 38.09%, while PLLaVA performs better at 59.24%, excelling in predictive and counterfactual reasoning among the baselines. Even with finetuning, InternVideo reaches only 52.62% , lagging behind our NS-4DPhysics. These results indicate that existing video foundation models struggle to reason effectively in complex dynamic scenes.

5.3 Analysis

We further analyze the effectiveness of two key components of NS-4DPhysics: the physics prior and the symbolic reasoning model. (1) Without the physics prior, the 4D world states are estimated using only the rendering likelihood, with time consistency maintained by initializing $\mathcal{S}_t$ from the previous frame, $\hat{\mathcal{S}}_{t-1}$. The result in Table 2 shows that adding the physics prior to the 4D scene representation significantly improves VideoQA performance. We also visualize the predicted world states of both models in Sec. 5.4 with the failure cases. (2) We also compare a variant of GPT4-o, where as a language model, it processes the predicted scene structure from NS-4DPhysics as text input instead of raw images. However, the LLM still underperforms in 4D dynamic reasoning, highlighting the advantages of our symbolic reasoning model in capturing dynamic object properties.

Table 2: Analysis of the physics prior module and symbolic reasoning (SR) in our model. We compare our variants without the physics prior and augment LLM with explicit 4D scene representation.

	Average	Factual				Predictive	Counterfactual
		All	Vel.	Acc.	Col.		
4D representation + SR (Ours)	**82.64**	**87.70**	**88.66**	**83.73**	**88.46**	**85.71**	**74.51**
w/o physics prior + SR	75.97	79.68	81.40	81.30	74.88	78.83	69.39
4D representation + GPT-4o	61.39	65.49	66.67	54.60	71.63	57.14	51.09
video + GPT-4o	56.06	55.50	58.81	57.67	47.12	56.93	58.16

5.4 Qualitative results

We visualize how our model achieves more accurate results in reconstructing scenes and simulating future scenes for predictive questions, assisted by the physics prior. Fig. 4 shows an example of a factual question. Without the physics prior (row c), the model fails to track the correct position of the minivan and provides the wrong answer regarding the collision.

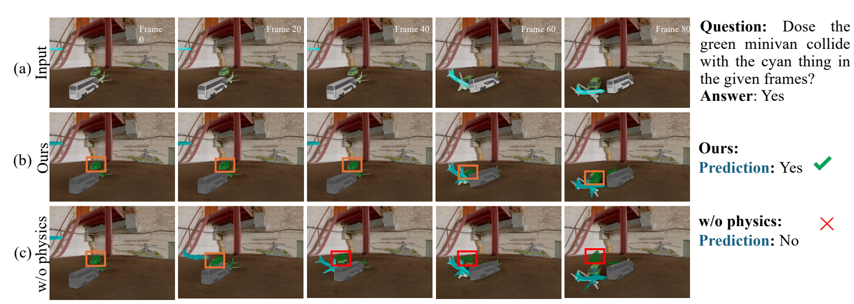


Figure 4: Qualitative examples of factual questions. (a) shows the input video; (b)We show our NS-4DPhysics can have a better estimation for the motion with the physical prior. (c) The error of position predicted by baseline w/o physics in the red box leads to the mistake in the answer.

Fig. 5 illustrates a predictive question. In row b1, our model predicts more accurate positions for all the objects, and the simulation of future frames in (b2) shows no collisions between the objects.

However, without the physics prior, the error in 3D position prediction (especially for the school bus) causes all the vehicles to collide in future frames.

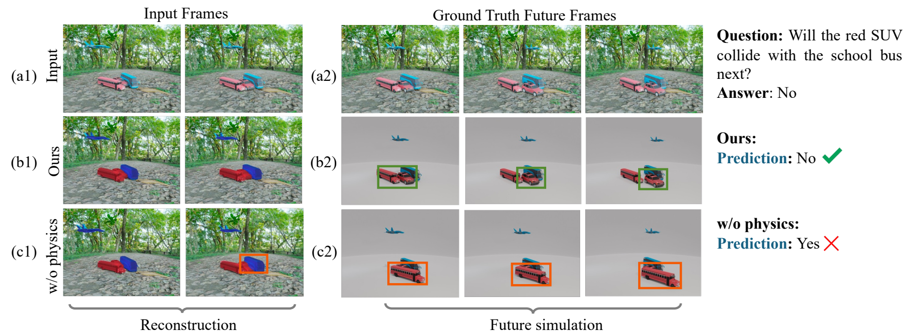
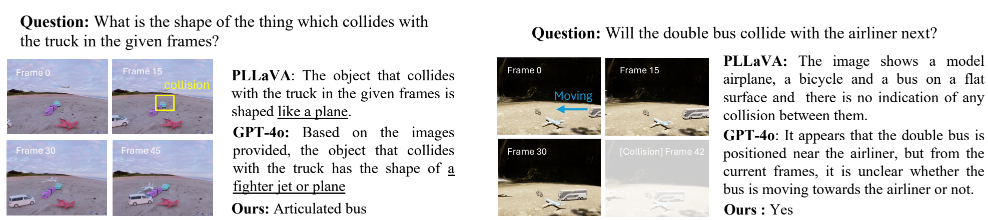


Figure 5: Qualitative examples of predictive questions. (a1) The first 30 frames are given to models as input video and (a2) the following are hidden as ground truth future states; (b1) our NS-4DPhysics has a better estimation of the poses of objects, and (b2) a plausible imagination by re-simulation. (c1) The red box shows the error of poses estimation of bus when without physics prior, which (c2) makes the red SUV collide with the school bus in the future.

Analysis of Video-LLM. To analyze the reasoning failures in VideoLLM, we examine two failure cases of PLLaVA and GPT-4o. (a) Misunderstanding the location and collisions in 3D space: Both PLLaVA and GPT-4o incorrectly predict that the truck collides with the fighter jet. In reality, the truck only collides with the purple articulated bus, while the plane is near the truck but does not make contact. This error is a result of relying on 2D projection, which fails to capture the correct 3D spatial relationships. (b) Fail to predict future 4D dynamics world states: *Both models identify the objects correctly*, but they fail to understand the objects' motion from the given frames. Although no collision occurs in the current frames, the bus is moving toward the airplane and will collide in the future—something the models are unable to predict.



(a) Misunderstand the location and collisions in 3D space (b) Fail to predict future 4D dynamics world states

Figure 6: Two failure cases of PLLaVA and GPT-4o. (a) Both models incorrectly predict a truck colliding with a fighter due to misunderstanding 3D spatial relationships, likely caused by reliance on 2D projection; (b) While the models correctly identify objects, they fail to predict future collisions as not understanding the 4D dynamics of objects from the given frames.

6 Conclusion and Discussion

In this work, we explore 3D dynamic properties in video question answering. We introduce DynSuperCLEVR to study 4D dynamics of scenes in the VideoQA tasks with improved realistic texture. Our findings show that existing video-language models, even those with large-scale pretraining, struggle to capture critical 4D dynamic properties necessary for temporal and predictive reasoning. To address this, we develop NS-4DPhysics, a neural-symbolic model that estimates an explicit 4D scene representation and answers questions through program execution. Experimental results on DynSuperCLEVR demonstrate that our approach significantly outperforms previous state-of-the-art methods, highlighting its effectiveness in inferring dynamic properties and predicting future states. In ongoing work, as Kaushik et al. (2024) has demonstrated the potential of 3D NMMs for domain adaptation from synthetic images to real image data, we are studying how our NS-4DPhysics can be extended to work with real images.

Limitations. As a synthetic VideoQA dataset, our DynSuperCLEVR currently focuses only on rigid objects with linear velocity and acceleration. More complex objects, such as articulated and non-rigid objects with complex trajectories, are not yet considered.

REFERENCES

Tayfun Ates, M. Ateşoğlu, Çağatay Yiğit, Ilker Kesen, Mert Kobas, Erkut Erdem, Aykut Erdem, Tilbe Goksun, and Deniz Yuret. CRAFT: A benchmark for causal reasoning about forces and inTeractions. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2602–2627, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.205. URL https://aclanthology.org/2022.findings-acl.205.

Yutong Bai, Angtian Wang, Adam Kortylewski, and Alan Yuille. Coke: Contrastive learning for robust keypoint detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 65–74, 2023.

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1728–1738, 2021.

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

Daniel M Bear, Elias Wang, Damian Mrowca, Felix J Binder, Hsiao-Yu Fish Tung, RT Pramod, Cameron Holdaway, Sirui Tao, Kevin Smith, Fan-Yun Sun, et al. Physion: Evaluating physical prediction from vision in humans and machines. *arXiv preprint arXiv:2106.08261*, 2021.

Zhenfang Chen, Kexin Yi, Yunzhu Li, Mingyu Ding, Antonio Torralba, Joshua B Tenenbaum, and Chuang Gan. Comphy: Compositional physical reasoning of objects and events from videos. *arXiv preprint arXiv:2205.01089*, 2022.

Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. 2016.

Mingyu Ding, Zhenfang Chen, Tao Du, Ping Luo, Josh Tenenbaum, and Chuang Gan. Dynamic visual reasoning by learning differentiable physics models from video and language. *Advances In Neural Information Processing Systems*, 34:887–899, 2021.

Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 961–970, 2015.

Rohit Girdhar and Deva Ramanan. CATER: A diagnostic dataset for Compositional Actions and TEmporal Reasoning. In *ICLR*, 2020.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, 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.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: a scalable dataset generator. 2022.

Jessica B Hamrick, Peter W Battaglia, Thomas L Griffiths, and Joshua B Tenenbaum. Inferring mass in complex scenes by mental simulation. *Cognition*, 157:61–76, 2016.

Artur Jesslen, Guofeng Zhang, Angtian Wang, Alan Yuille, and Adam Kortylewski. Robust 3d-aware object classification via discriminative render-and-compare. *arXiv preprint arXiv:2305.14668*, 2023.

Achuta Kadambi, Celso de Melo, Cho-Jui Hsieh, Mani Srivastava, and Stefano Soatto. Incorporating physics into data-driven computer vision. *Nature Machine Intelligence*, 5(6):572–580, 2023.

Prakhar Kaushik, Aayush Mishra, Adam Kortylewski, and Alan Yuille. Source-free and image-only unsupervised domain adaptation for category level object pose estimation. *arXiv preprint arXiv:2401.10848*, 2024.

Nikolaos Kyriazis and Antonis Argyros. Physically plausible 3d scene tracking: The single actor hypothesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9–16, 2013.

Zhuowan Li, Xingrui Wang, Elias Stengel-Eskin, Adam Kortylewski, Wufei Ma, Benjamin Van Durme, and Alan L Yuille. Super-clevr: A virtual benchmark to diagnose domain robustness in visual reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14963–14973, 2023.

Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023.

Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7708–7717, 2019.

Wufei Ma, Angtian Wang, Alan Yuille, and Adam Kortylewski. Robust category-level 6d pose estimation with coarse-to-fine rendering of neural features. In *European Conference on Computer Vision*, pp. 492–508. Springer, 2022.

Wufei Ma, Kai Li, Zhongshi Jiang, Moustafa Meshry, Qihao Liu, Huiyu Wang, Christian Häne, and Alan Yuille. Rethinking video-text understanding: Retrieval from counterfactually augmented data. *arXiv preprint arXiv:2407.13094*, 2024.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, et al. OpenEQA: Embodied Question Answering in the Era of Foundation Models. In *CVPR*, 2024.

Maitreya Patel, Tejas Gokhale, Chitta Baral, and Yezhou Yang. Cripp-vqa: Counterfactual reasoning about implicit physical properties via video question answering. *arXiv preprint arXiv:2211.03779*, 2022.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

Mathieu Salzmann and Raquel Urtasun. Physically-based motion models for 3d tracking: A convex formulation. In *2011 International Conference on Computer Vision*, pp. 2064–2071, 2011. doi: 10.1109/ICCV.2011.6126480.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

Fish Tung, Mingyu Ding, Zhenfang Chen, Daniel M. Bear, Chuang Gan, Joshua B. Tenenbaum, Daniel L. K. Yamins, Judith Fan, and Kevin A. Smith. Physion++: Evaluating physical scene understanding that requires online inference of different physical properties. *arXiv*, 2023.

Tomer D Ullman, Andreas Stuhlmüller, Noah D Goodman, and Joshua B Tenenbaum. Learning physical parameters from dynamic scenes. *Cognitive psychology*, 104:57–82, 2018.

Angtian Wang, Adam Kortylewski, and Alan Yuille. Nemo: Neural mesh models of contrastive features for robust 3d pose estimation. *arXiv preprint arXiv:2101.12378*, 2021.

Xingrui Wang, Wufei Ma, Zhuowan Li, Adam Kortylewski, and Alan Yuille. 3d-aware visual question answering about parts, poses and occlusions. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL `https://openreview.net/forum?id=AMIJEupsNq`.

Xingrui Wang, Wufei Ma, Zhuowan Li, Adam Kortylewski, and Alan L Yuille. 3d-aware visual question answering about parts, poses and occlusions. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 58717–58735. Curran Associates, Inc., 2023b. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/b783c44ba9adbc30344473dc633b4869-Paper-Conference.pdf`.

Xingrui Wang, Wufei Ma, Zhuowan Li, Adam Kortylewski, and Alan L Yuille. 3d-aware visual question answering about parts, poses and occlusions. *Advances in Neural Information Processing Systems*, 36, 2024.

Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022.

Bo Wu, Shoubin Yu, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan. Star: A benchmark for situated reasoning in real-world videos. *arXiv preprint arXiv:2405.09711*, 2024.

Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. *Advances in neural information processing systems*, 28, 2015.

Jiajun Wu, Erika Lu, Pushmeet Kohli, Bill Freeman, and Josh Tenenbaum. Learning to see physics via visual de-animation. *Advances in neural information procesing systems*, 30, 2017.

Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*, 2024.

Hongyang Xue, Zhou Zhao, and Deng Cai. Unifying the video and question attentions for open-ended video question answering. *IEEE Transactions on Image Processing*, 26(12):5656–5666, 2017. doi: 10.1109/TIP.2017.2746267.

Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Learning to answer visual questions from web videos. *IEEE TPAMI*, 2022.

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. Clevrer: Collision events for video representation and reasoning. *arXiv preprint arXiv:1910.01442*, 2019.

Zhicheng Zheng, Xin Yan, Zhenfang Chen, Jingzhou Wang, Qin Zhi Eddie Lim, Joshua B Tenenbaum, and Chuang Gan. Contphy: Continuum physical concept learning and reasoning from videos. *arXiv preprint arXiv:2402.06119*, 2024.

A 3D OBJECTS AND COLORS

In DynSuperCLEVR, we have 21 object classes and 8 colors. As described in the main paper, we improve upon previous datasets by generating new 3D textures with different colors. Below, we present all the 3D objects, with one object selected in each color.

Figure 7: Examples of 3D objects in DynSuperCLEVR with a selected color for each object.

B DYNAMIC PROPERTIES SETTING

When constructing DynSuperCLEVR, we need to set the extra physical properties to objects for physics simulation. In the main paper, we have introduced the design of the 4D dynamics properties like position, velocity and acceleration. Here we describe how the exact values are set in the physics engine in Table. 3. Also, we need to set a series of physical properties in the physics engine, such as mass, friction and restitution.

(1) **Mass and gravity:** The mass for each object is calculated based on its specific shape model, which is linearly related to its volume with a density of 2.7; All objects in the dataset are subject to gravity, which influences their vertical movements and impacts when in flight or during falls. The gravitational constant is set as 10.

(2) **Friction**: Frictional forces affect the movement of all objects, especially when they interact with the ground or each other, slowing them down and eventually bringing them to a stop. The friction of objects are set as 0.2; the friction floor (the dome shaped background) is set as 0.4.

(3) **Restitution**: When objects collide or drop to the ground, the elastic forces come into play, defined by elasticity coefficients. These forces affect how objects bounce off each other or rebound from barriers, significantly altering their trajectories and speeds. The restitution of objects and the floor's are set to 0.5.

Table 3: Dynamic settings of DynSuperCLEVR

Attribute	**Description**	**Aeroplanes**	**Others**
Mass	Calculated from volume	ρV	
Position	(x, y)	Beta distribution	
	z	Uniform distribution	0
Orientation	Faces to the center with noises	-	
Velocity	Initial state: static, slow, fast	$\{0, 3, 6\}$ m/s	
Internal Force	Engine force (Forwards)	$\{1, 0\}$m/s^2	
	Floating force(Upwards)	{10, 0} Mass $\times$ m/s^2	0 Mass $\times$ m/s^2

C PROGRAM EXECUTION

In our DynSuperCLEVR, we introduce new programs for the 4D dynamics properties and the collisions events. In table 4, we list all the programs and their input / output types, involved in the DynSuperCLEVR.

Table 4: All operations and their input / output types involved in the DynSuperCLEVR

Type	Operation	Input Type	Output Type
Event Operations	`filter_collision`	CollisionEventSet, Object	CollisionEventSet
	`get_all_col_partners`	CollisionEventSet, Object	ObjectSet
	`get_frame`	CollisionEvent	FrameID
	`come_in_frame`	Object	FrameID
Object filter Operations	`filter_attributes`	ObjectSet	ObjectSet
	`filter_static`	ObjectSet, FrameID	ObjectSet
	`filter_moving_velocity`	ObjectSet, FrameID	ObjectSet
	`filter_accelerating`	ObjectSet, FrameID	ObjectSet
	`filter_floating`	ObjectSet, FrameID	ObjectSet
Object Query Operations	`query_attributes`	Object	Shape or color
	`is_static`	Object, FrameID	Bool
	`query_moving_velocity`	Object, FrameID	Velocity
	`query_moving_direction`	Object, FrameID	Direction
	`is_accelerating`	Object, FrameID	Bool
	`is_floating`	Object, FrameID	Bool
Object comparison	`faster_velocity`	Object, Object, FrameID	Bool
	`slower_velocity`	Object, Object, FrameID	Bool
Input Operations	`Objects`	None	ObjectSet
	`Events`	None	CollisionEventSet
	`futureEvents`	None	CollisionEventSet
Counterfactual Operation	`Counterfactual_static`	Object	CollisionEventSet
	`Counterfactual_moving_slow`	Object	CollisionEventSet
	`Counterfactual_moving_fast`	Object	CollisionEventSet
	`Counterfactual_accelerating`	Object	CollisionEventSet
	`Counterfactual_floating`	Object	CollisionEventSet
Others	`unique`	ObjectSet	Object
	`exist`	ObjectSet	Bool

Based on the programs above, we generate the corresponding program execution for each factual, predictive and counterfactual question. Here we provide more example of the program execution in Fig. 8.

D DETAILS OF 3D NNMS SCENE PARSER WITH PHYSICS PRIOR

The ultimate goal of the scene parser is to estimate the all dynamic state $\mathcal{S}_t$ from the corresponding observation $\mathcal{I}_t$ and the previous states $\mathcal{S}_{<t}$ using a probabilistic model. The previous work in 6D pose estimation has proposed to use render-and-compare to maximize scene likelihood for static images. For the video data, we further consider the temporal consistency and physical plausibility of the dynamics estimation.

$$\hat{\mathcal{S}}_t = \arg\max_{\mathcal{S}_t} p\left(f(\mathcal{I}_t)|\mathcal{S}_t\right) \cdot p(\mathcal{S}_t|\mathcal{S}_{<t}), \tag{5}$$

$$\hat{\mathcal{S}}_0 = \arg\max_{\mathcal{S}_0} p(f(\mathcal{I}_0)|\mathcal{S}_0). \tag{6}$$

D.1 3D NEURAL MESH MODEL

Inspired by the 3D-neural-meshed based generative mode for pose estimation methods on static images Wang et al. (2021); Ma et al. (2022), we develop a de-rendering based generative model to

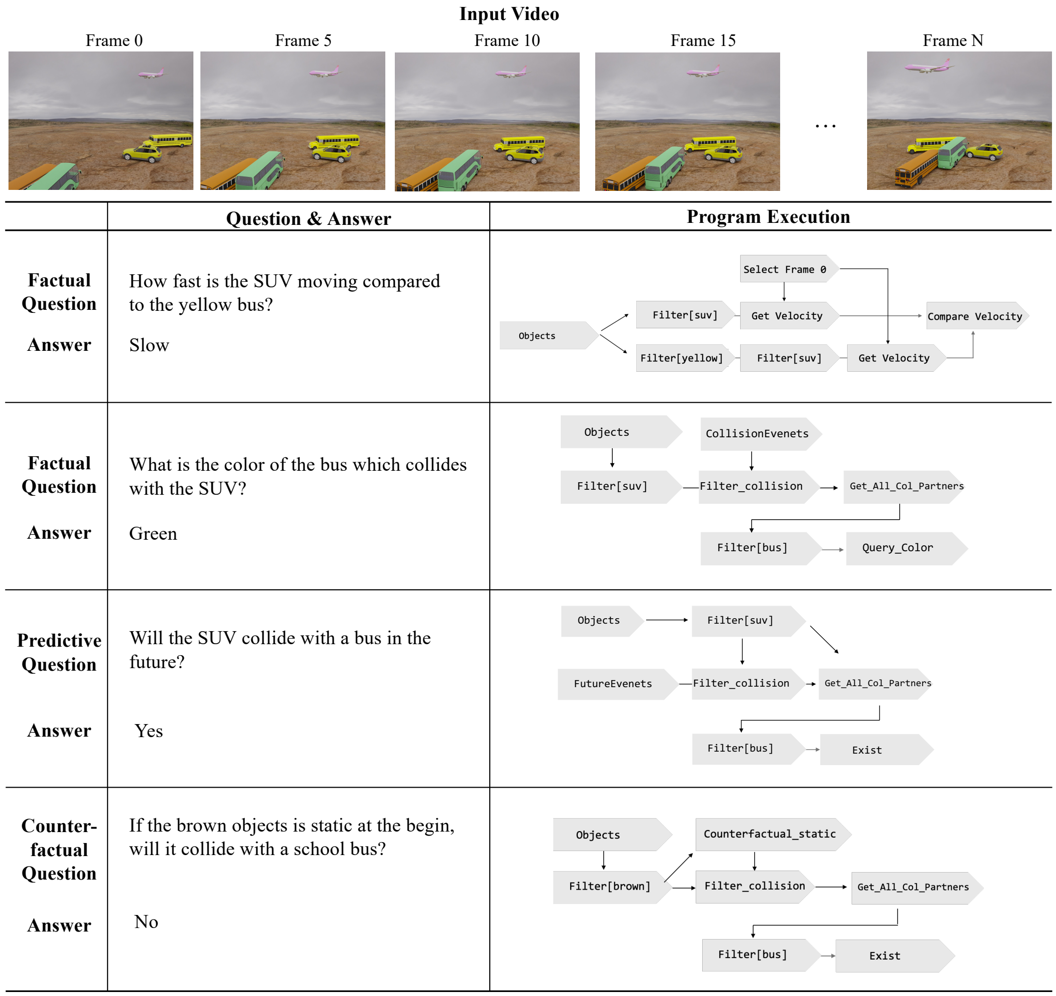


Figure 8: Examples of the reasoning program. From the input questions, we provide new operation programs to answer these questions step by step using a 4D dynamic scene representation defined in the main paper.

reconstruct the 4D dynamic scene representation frame by frame. To ensure the predictions are not only plausible in 3D positions but also in physical rules, we incorporate a physical likelihood model and a 3D generative model with rendering likelihood, as shown in Fig. 3 in the main paper. Here we describe the more details of the 3D neural mesh models as preliminary.

Preliminaries. In the previous work for static imagesMa et al. (2022), Neural Meshes model were introduced for 6D pose estimation through inverse rendering. For that task, the goal is to jointly estimate the 6D pose (2D location (x, y), distance d to the camera and 3D pose (α, β, γ) of objects in an image by comparing the Neural Meshes feature after rendering with the input image feature and maximizing the rendering likelihood. More formally, the mesh for a given object in category c is represented as $M_c = \{v_i \in \mathbb{R}^3 | i = 1 \dots N\}$ where v_i means the vertex. The corresponding neural texture of the mesh M_c is $T_c \in \mathbb{R}^{N \times l}$ where l is the dimension of the feature. So that the neural mesh model for category c is their aggregation $O_c = \{M_c, T_c\}$. The render-and-compare process as an optimization of the likelihood model:

$$p(F \mid O_c, \alpha_s, B) = \prod_{i \in \mathcal{FG}} p(f_i \mid O_c, \alpha_c) \prod_{i \in \mathcal{BG}} p(f_i' \mid B) \tag{7}$$

where $\mathcal{FG}$ and $\mathcal{BG}$ are the set of foreground and background locations on the 2D feature map and f_i is the feature vector of F at location i, produced by CNN feature extractor Φ. Here the foreground and background likelihoods are modeled as Gaussian distributions.

In this work, we use transform the position into a world coordinate instead. Given the object 3D rotation $R = (\alpha, \beta, \gamma)$ and translation $T = (x, y, z)$, we can render the neural mesh model O_c into a feature map F_c with soft rasterization Liu et al. (2019):

Training: During training, we follow (Ma et al., 2022) and first jointly train a CNN feature extractor Φ, the neural texture $\{T_y\}$ and the background model B. We utilize the EM-type learning strategy as originally introduced for keypoint detection in CoKe(Bai et al., 2023). Specifically, the feature extractor that produces f_i is trained using stochastic gradient descent while the parameters of the generative model $\{T_y\}$ and B are trained using momentum update after every gradient step in the feature extractor, which was found to stabilize training convergence. The final loss function is defined as a constructive loss between the feature of vertex.

$$\mathcal{L}_{\text{contrastive}} = -\sum_{i \in \mathcal{FG}} \sum_{i \in \mathcal{FG} \setminus \{i\}} \|f_i - f_j\|^2 - \sum_{i \in \mathcal{FG}} \sum_{i \in \mathcal{BG}} \|f_i - f_j\|^2 \tag{8}$$

Inference as scene parsing For static images, the previous work Wang et al. (2021) has shown that the rendering likelihood can be used to estimate the 6D pose of objects. In our dynamics scene, similar strategies can be applied to each frame at time step t. As the neural mesh models are probabilistic generative model of neural feature activation, we can first define the rendering likelihood of the feature map F_t given any 6D pose R_t, T_t as:

$$p(\mathcal{I}_t|\mathcal{S}_t) = p(F_t \mid O_c, R_t, T_t, B) \tag{9}$$

$$= \prod_{i \in \mathcal{FG}} p(f_t^{(i)} \mid O_c, R_t, T_t) \prod_{j \in \mathcal{BG}} p(f_t^{(j)} \mid B), \tag{10}$$

where $\mathcal{FG}$ and $\mathcal{BG}$ are the set of foreground and background locations on the 2D feature map and $f_t^{(i)}$ is the feature vector of F at location i at timestep t. B is a background parameter learned from training. Here the foreground and background likelihoods are modeled as Gaussian distributions.

D.2 Physical prior

Rendering likelihood alone is insufficient to reconstruct a physically plausible 4D dynamic scene representation. We also integrate a physics prior into our likelihood model. The physical process can be modeled as a Markov model Salzmann & Urtasun (2011), where the physical prior distribution of rotation R_t and translation T_t at time step t can be expressed as:

$$p(\mathcal{S}_t|\mathcal{S}_{<t}) = q(R_t, T_t \mid \hat{R}_{t-1}, \hat{T}_{t-1}). \tag{11}$$

However, it's nontrivial to directly model the physical functions. Most current studies on physical engines within neural networks only focus on simple shapes like cubes as the complexity in shapes of objects and the interaction process is hard to model. On the other hand, computational physical engines such as Bullet excel at modeling physical functions for any known shapes but are not differentiable and challenging to integrate into the de-rendering process during inference.

In our method, we modify the discriminative physics engineering into a probabilistic model by introducing an uncertainty term $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. Denoted a physics engine as PE$(\cdot)$, then the $q(R_t, T_t \mid \hat{R}_{t-1}, \hat{T}_{t-1})$ can be assumed by:

$$(R_t, T_t) = \text{PE}(\hat{R}_{t-1}, \hat{T}_{t-1}) + \varepsilon, \tag{12}$$

$$(R_t, T_t) \mid (\hat{R}_{t-1}, \hat{T}_{t-1}) \sim \mathcal{N}(\text{PE}(\hat{R}_{t-1}, \hat{T}_{t-1}), \sigma^2 I), \tag{13}$$

$$q(R_t, T_t \mid \hat{R}_{t-1}, \hat{T}_{t-1}) = C \exp\left(-\frac{1}{2\sigma^2} \left[(R_t, T_t) - \mu\right]^T \left[(R_t, T_t) - \mu\right]\right), \tag{14}$$

where $\mu = \text{PE}(\hat{R}_{t-1}, \hat{T}_{t-1})$, $C = 1/\sqrt{(2\pi)^k |\sigma^2 I|}$.

Finally, the rotation R_t and translation T_t can be estimated by maximizing the joint likelihood of the rendering likelihood and the physical prior likelihood:

$$\hat{R}_t, \hat{T}_t = \arg\max_{R_t, T_t} p(F_t \mid O_c, R_t, T_t, B) \cdot q(R_t, T_t \mid \hat{R}_{t-1}, \hat{T}_{t-1}). \tag{15}$$

Relationship to Bayes-Kalman filtering The conceptual ideas are directly inspired by the classic ideas of Bayes-Kalman filtering where the goal is to estimate a hidden state based on a sequence of observations. Bayes-Kalman filtering consists of updating a probability distribution of the hidden state by a prediction step followed by a correction step which incorporates evidence from a new observation. It uses a dynamic model for how the hidden state changes with time which is directly analogous to our physical prior. It has an observation model, corresponding to our scene parser, for how new observations give evidence for the hidden state. Bayes-Kalman, however, is difficult to implement in complex applications, like ours, because it requires us to represent and update a complex probability distribution. The standard approach for doing this is particle filtering where the probability distribution is represented by a set of point particles which are updated during the prediction and correction steps. This is also challenging so instead we use a simple approximation which essentially uses a single particle. In future work we will experiments to see if our model gives even better results if instead of this approximation we use particle filtering.

E DETAILS OF THE BASELINE MODELS

We select 7 representative models from the following 3 categories as baseline models on DynSuper-CLEVR.

(1) Simple classification-based methods encode videos with a CNN backbone and predict answers with a classifier head. Specifically, we consider ***CNN+LSTM***, which aggregates frame-level CNN features and encodes the question using an LSTM, and ***FiLM*** Perez et al. (2018), which incorporates a feature-level linear modulation module for question answering.

(2) Neural symbolic models first parse the scene into object instances and then execute a program for question answering. ***NS-DR*** Yi et al. (2019) adopts Mask R-CNN for object detection. We modify it with the new program in DynSuperCLEVR and evaluate it on factual questions. Since their 2D simulator is unable to reason about objects in 3D, we do not compare it on predictive and counterfactual questions. ***PO3D-VQA*** Wang et al. (2024) uses a 3D detector and reconstructs an explicit 3D scene representation for each frame. We extend the model for VideoQA by computing dynamic properties from object locations and predicting collisions by filtering the distance between objects.

(3) Large multimodal models leverage large-scale image-text or video-text data for pretraining and achieve strong generalization abilities across various video-text tasks. We consider ***Video-LLaVA*** Lin et al. (2023) and ***PLLaVA*** Xu et al. (2024) for zero-shot evaluation, where a GPT model is used to evaluate the correctness of free-form answers. Additionally, we fine-tune a pretrained ***InternVideo*** Wang et al. (2022) model that predicts answers using a classifier head. Finally, we evaluate the proprietary model **GPT-4o** by accessing it through the OpenAI API with customized system prompts. Specifically, we consider two settings: (i) asking the GPT-4o model to treat the problem as a multiple-choice question and predict the answer directly (see Figure 9); and (ii) asking the GPT-4o model to think step by step, provide necessary reasoning about the dynamic and physical events, and then give the answer to the question (see Figure 10).

Zero-shot evaluation of GPT-4o with multiple choice questions

System:
You are an intelligent chatbot designed for answering questions based on eight frames obtained from a video.

Your task is to analyze the frames of the video, identify the nature of the object movements (such as speed, acceleration, and direction), and then determine the answer to the question.

INSTRUCTIONS:
Directly answer the question with one of the choices: airliner, articulated bus, back, blue, brown, chopper motorcycle, cruiser, cyan, dirtbike, double bus, down, false, fighter aircraft, front, gray, green, jet, left, minivan, mountain bike, purple, red, right, school bus, scooter, sedan, suv, tandem bike, truck, true, up, utility bike, wagon, yellow.

User:
Watch the eight frames in the video and answer the question: Is the red mountain moving fast at the beginning?

GPT-4o:
false

Figure 9: **Zero-shot evaluation of GPT-4v with multiple choice questions.** We evaluate the proprietary model GPT-4o by accessing the model through OpenAI API with customized system prompts.

Zero-shot evaluation of GPT-4o with step-by-step reasoning

System:
You are an intelligent chatbot designed for answering questions based on eight frames obtained from a video.

Your task is to analyze the frames of the video, identify the nature of the object movements (such as speed, acceleration, and direction), and then determine the answer to the question.

INSTRUCTIONS:
- Think step by step, reason about the dynamic and physical events, and then give the answer.
- End your response with the exact answer to the question, from one of the choices: airliner, articulated bus, back, blue, brown, chopper motorcycle, cruiser, cyan, dirtbike, double bus, down, false, fighter aircraft, front, gray, green, jet, left, minivan, mountain bike, purple, red, right, school bus, scooter, sedan, suv, tandem bike, truck, true, up, utility bike, wagon, yellow.

User:
Watch the eight frames in the video. Think step by step and then answer the question: Is the red mountain moving fast at the beginning?

GPT-4o:
To determine if the red mountain bike is moving fast at the beginning of the sequence, we need to closely analyze its position in the first few frames:

1. In the first frame, observe the position of the red mountain bike.
2. In the second frame, note any change in position of the red mountain bike.
3. Compare the distance covered by the red mountain bike between these frames.

If the red mountain bike has moved a significant distance between the first two frames, it is moving fast at the beginning. If it has moved only a small distance, it is not moving fast.

From the frames:

- The red mountain bike does not show significant movement between the first two frames.
- The red mountain bike appears to be in almost the same position.

Thus, the red mountain bike does not move fast at the beginning.

The answer is: false.

Figure 10: **Zero-shot evaluation of GPT-4v with step-by-step reasoning.** We evaluate the proprietary model GPT-4V by accessing the model through OpenAI API with customized system prompts. Specifically, we encourage the GPT-4o model to think step by step and reason about the dynamic and physical events before predicting the answer.