Goal Force: Teaching Video Models To Accomplish Physics-Conditioned Goals

Supplementary Material

7. Additional Experiment Details

7.1. Comparison to Prior Works: Direct Force Prompting Quantitative Comparison

We encode the goal force prompt in the second channel of the control signal, and we encode the direct force prompt (which is a similar task to PhysDreamer [63], Force Prompting [17], and PhysGen [37]) in the first channel of the control signal. In Table 4 we compare the "Direct Force Prompting" capability of our model to those three models via a 2AFC human study ($N = 10$) conducted on Prolific. We gathered two benchmarks: a PhysGen benchmark, consisting of four scenes highlighted on that work's project page; as well as a PhysDreamer benchmark, consisting of three scenes highlighted on that work's project page. We compare our model to PhysGen and Force Prompting on the PhysGen benchmark, and we compare our model to PhysDreamer and Force Prompting on the PhysDreamer benchmark. Note that PhysGen models rigid body mechanics, whereas PhysDreamer models oscillations.

7.2. Synthetic Data Generation

In this section, we provide an in-depth discussion of the methods and specific parameters utilized for generating our synthetic training data. This data is used to train the Goal Force model to act as an implicit neural physics planner.

For all synthetic datasets, a key step in creating the multi-channel control signal $\tilde{\pi}$ is the projection of 3D forces and object properties onto the 2D image plane. We use the camera's parameters to map 3D force vectors and object world coordinates into 2D pixel coordinates, enabling us to accurately model physical interactions within the video frames.

7.2.1. Dominos Dataset

We generated 3k videos of domino chain reactions using Blender. The setup models a causal chain where an initial direct force on one domino results in a predictable goal force on a downstream target domino.

To ensure diversity and robustness, we randomized the following parameters per video:

- **Domino Count:** Uniformly sampled from $\mathrm{Unif}\{3, \ldots, 10\}$.
- **Scene Geometry:** Randomized placement and orientation of the domino line.
- **Causality:** Choice of the initial target domino and the direction of the chain reaction (i.e., hitting the domino in front or behind).
- **Visuals:** Randomized camera position, ground textures (from 42 Polyhaven options), and High Dynamic Range Images (HDRIs) for lighting and background (from 50 Polyhaven options).
- **Force Magnitude:** Continuous values from $[0, 1]$, where 0 represents the minimum force required for the domino to topple and 1 represents a maximal, strong impulse.

Each video is accompanied by a JSON file that records the names of the initial and adjacent contact dominos, along with the complete 2D pixel coordinates for all dominos across every frame.

7.2.2. Rolling Balls Dataset

This dataset comprises 6k videos generated in Blender, split into two primary categories to capture both collision and non-collision causal interactions:

1. Collision Set (4.5k videos): A "projectile" ball, acted upon by an unseen point force, is aimed to collide with one specific "target" ball within a group of initially stationary "distractor" balls.
2. Non-Collision Set (1.5k videos): The projectile ball is aimed such that it misses the target ball.

For the Collision Set, we ensured a diverse range of physical scenarios by randomizing:

- **Ball Count:** $\mathrm{Unif}\{3, \ldots, 9\}$.
- **Physical Properties:** Ball colors, ball masses $\mathrm{Unif}(1.0, 4.0)$ kg, and all ball positions.
- **Visuals:** Randomized camera position and ground textures.
- **Force Calculation:** To guarantee collision, a minimum required force is calculated based on the projectile mass, distance to the target, and a randomized collision time ($\mathrm{Unif}(2.5, 4.5)$ seconds). This minimum force is scaled by $\mathrm{Unif}(1.2, 1.6)$ to introduce physical variation.

The collision videos are evenly split between straight-on and indirect collisions. For both types, the script first calculates the precise angular window necessary for the projectile to hit the target.

- For straight-on collisions, the force is aimed directly at the center of this calculated angular window.
- For indirect collisions, the force angle is randomly sampled within this window, resulting in an off-center strike.

This mixed-collision approach helps the model learn diverse post-collision behaviors.

For the Non-Collision Set, we randomized: ball quantity ($\mathrm{Unif}\{3, \ldots, 5\}$), ball textures, positions, camera angle, ground textures, target ball selection, force angle ($[0, 360°)$), and force magnitude ($[0, 1]$).

For all ball videos, a JSON file records initial 2D/3D coordinates and physics parameters. For the videos featuring indirect collisions, we also save the complete 2D pixel tra-

Visual Quality	Dominos	Pool balls	Stone tower	Wall toy	Orange Rose	White Rose	Tulip
Force Prompting	90.0%	80.0%	60.0%	50.0%	100.0%	80.0%	60.0%
PhysGen	60.0%	100.0%	100.0%	80.0%	–	–	–
PhysDreamer	–	–	–	–	50.0%	50.0%	50.0%

Force Adherence	Dominos	Pool balls	Stone tower	Wall toy	Orange Rose	White Rose	Tulip
Force Prompting	90.0%	90.0%	80.0%	90.0%	50.0%	60.0%	50.0%
PhysGen	90.0%	80.0%	80.0%	40.0%	–	–	–
PhysDreamer	–	–	–	–	40.0%	30.0%	60.0%

Table 4. **Human study comparing the Direct Force capability of the Goal Force method to prior works.** Numbers indicate the percentage of human pairwise preferences for Goal Force Prompting's direct force capability (i.e. encoding the force in the first channel) over each baseline on each benchmark dataset. The results demonstrate that Goal Force achieves consistently higher visual quality, as well as superior force adherence against the majority of baselines. Notably, our method achieves these results without relying on physics simulators or 3D assets at inference, unlike PhysDreamer and PhysGen. We note that PhysGen models rigid body mechanics, whereas PhysDreamer models oscillations, so they can't be directly compared to one another.

Scene	**# Valid**	**# Success**	**% Accuracy**
Dominos	50	50	100.00
Pool Scene 1	49	48	97.96
Pool Scene 2	22	12	54.55
Duckie Scene 1	40	34	85.00
Duckie Scene 2	37	24	64.86
Duckie Scene 3	41	38	92.68
Kitchen Lemon	50	50	100.00
Kitchen Cantaloupes	50	39	78.00
Paper Balls	50	49	98.00
Coffee Cups	44	41	93.18
Accessories	50	47	94.00
Curling Stones	49	37	75.51
Rubik's Cube	49	46	93.88
Curling Stones - Tool Use	47	45	95.74
Accessories - Tool Use	47	40	85.11
Coffee Cups - Tool Use	44	44	100.00
Air Hockey - Tool Use	45	38	84.44
Kitchen Lemon - Tool Use	50	44	88.00
Paper Balls - Tool Use	50	47	94.00
Plant Pots - Tool Use	46	38	82.61
Soaps - Tool Use	48	42	87.50
Rubik's Cube - Tool Use	48	45	93.75

Table 5. **Visual planning results for all test scenes.** We also report the visual planning accuracy for diverse tool use scenarios, where success is defined as correctly using tools to achieve the specified goal force in the presence of distractors. The results show that the model achieves a high success rate in most cases across diverse and complex scenarios.

jectory of the target ball. For the non-collision videos, we save the final 2D trajectory angle of the projectile ball.

7.2.3. Plants Dataset

This dataset, generated using PhysDreamer [63] (which integrates 3D Gaussians and a physics simulator), focuses on non-rigid body dynamics. The videos show a plant (carnation) swaying after being subjected to a direct force. We randomized the following parameters:

- **Initial Conditions:** Camera position and initial object configuration.
- **Force Application:** Contact points, force angles, and force magnitudes in $[0, 1]$, where 0 is a gentle poke and 1 is a strong impulse.

7.3. Ablation studies

7.3.1. How important is the mass channel?

We find that masking the mass channel during training and relying on text instead for mass leads to worse performance than reported in Figure 6.

7.3.2. How important is the direct force channel?

We find that masking the direct force channel during training causes the model to fail on some of the complex, out-of-domain causal chains, such as human-object interactions.

7.3.3. How specific does the text prompt need to be?

We conducted ablation studies to confirm that the action itself does not need to be specified in the text prompt (*e.g.*, "the dog paw *causes a ball to move*" works just as well as "the dog paw *nudges the ball*"). When the source of action is unspecified, the model resorts to its pre-trained prior to choose a cause, which may sometimes be an "invisible" force. We highlight that, even when the source of action is specified in the text prompt, the low-level physical dynamics are not (*e.g.*, we never specify to "hit the ball at that angle with that force"). Additionally, the ability to specify the interaction source and type is a desirable feature (*e.g.*, for robotics applications), as it allows users to provide the embodiment and the high-level plan information.

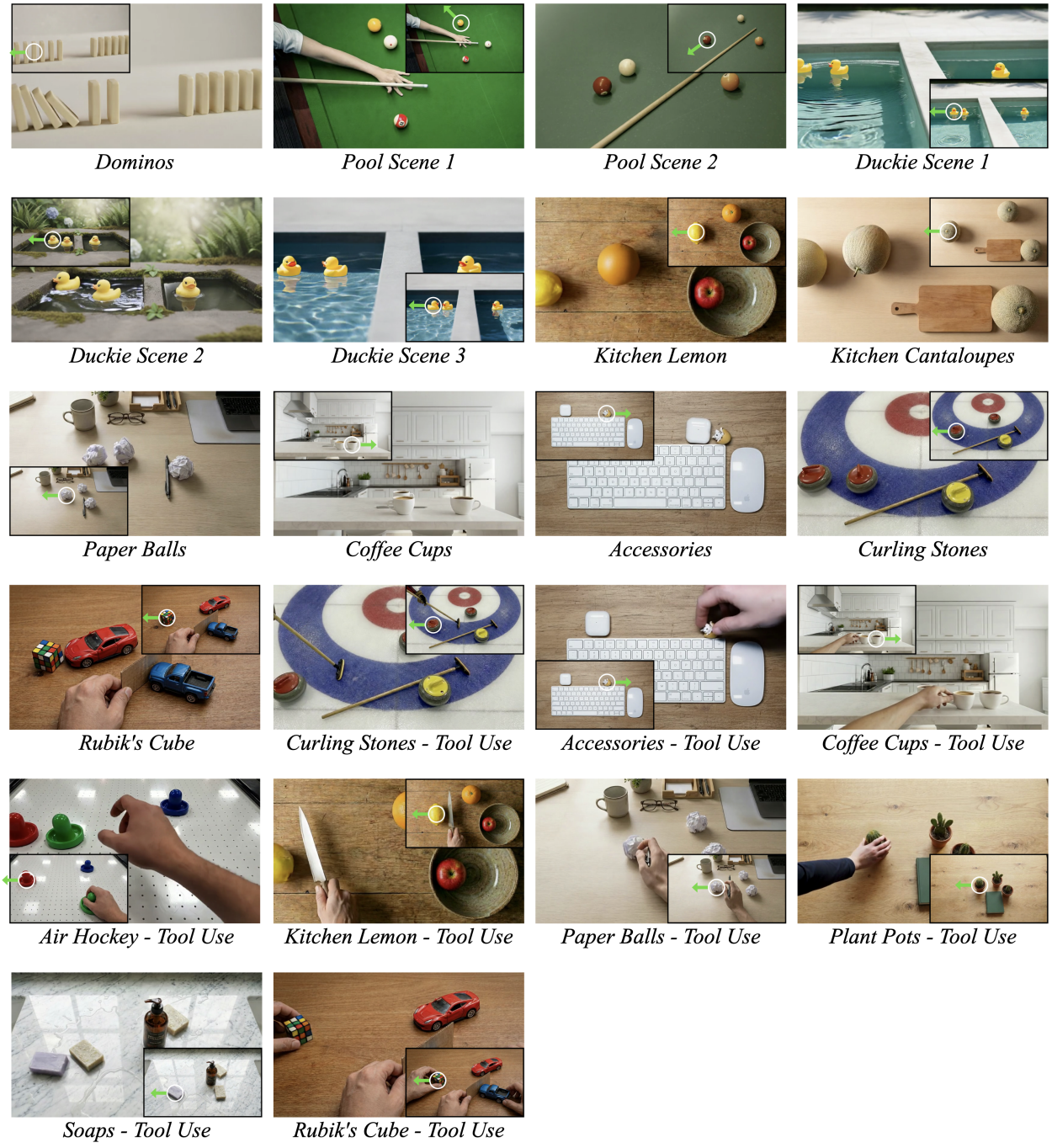

Figure 7. Visualization of all visual planning test scenes. The inset shows the initial state, where the green arrow indicates the goal force, while the larger image shows a valid outcome.