PhysVid: Physics Aware Local Conditioning for Generative Video Models

Supplementary Material

6. Additional Information

We begin with preliminaries, wherein we provide a brief overview of RoPE followed by a procedural description of the chunk aware cross-attention mechanism to augment the discussion in Sec. 3. Subsequently, we supplement Sec. 5 with a discussion of works closely related to PhysVid followed by a discussion on its limitations and future opportunities for contribution.

6.1. Rotary Positional Embeddings (RoPE)

RoPE is an established method for encoding positional information within transformer-based models that uniquely captures both absolute and relative positional data through vector rotations [44]. Fundamentally, the goal is to apply a set of block-diagonal rotation matrices R to the query vectors q and the key vectors k at each position. Thus, the transformations for the d dimensional query vector q_m and the key vector k_n at the positions m and n, respectively, are

$$q'_m = R_m q_m \qquad k'_n = R_n k_n \tag{3}$$

where R_m and R_n are the corresponding block-diagonal rotation matrices consisting of $d/2$ blocks. Each diagonal block $R_{m,i}$ in R_m corresponds to dimensions $2i-1, 2i$ and is defined as

$$R_{m,i} = \begin{pmatrix} \cos(m\theta_i) & -\sin(m\theta_i) \\ \sin(m\theta_i) & \cos(m\theta_i) \end{pmatrix} \tag{4}$$

where θ_i is a predefined frequency term. Since $R_m^T R_n = R_{n-m}$, this design ensures that the inner product of the rotated query and key vectors $(q'_m)^T k'_n$ depends only on the original query and key vectors and their relative distance, $n-m$.

6.2. Chunk aware cross attention procedure

In PhysVid, the chunk-wise positional alignment between visual query tokens and textual key tokens within the local pathway is achieved through a coherent grid-based encoding scheme. The procedure is described in Algorithm 1. Specifically, the text tokens from all video segments are concatenated and introduced into the local attention pathways as illustrated in Fig. 3. To preserve and utilize the contextual position of each token, a two-dimensional coordinate grid is imposed on the set of text key tokens. Within this grid, the first dimension indexes the corresponding video chunk, while the second dimension identifies the intra-chunk position. RoPE is then applied to encode these 2D coordinates in the representation of each key token. This design ensures that global and local positional information is preserved for each token throughout the network. Thus, the subsequent cross-attention mechanism can attend to localized content across all chunks, while maintaining precise chunk-specific referencing and temporal awareness.

Algorithm 1 Chunk Aware Cross-Attention

Require: Video tokens $X \in \mathbb{R}^{B \times L_v \times H \times d}$ ▷ L_v: video sequence length
Require: Local text representations $\{T^{(b)}\}_{b=1}^{N_b}$ for N_b video chunks
Require: Video grid $G_v \in \mathbb{N}^{B \times 3}$, RoPE frequencies Ω
Require: Number of chunks N_b, per-chunk text length L_c
Ensure: Updated video representation $\hat{X}$ after chunk aware cross-attention

// 1. Concatenate local text across all chunks
$T \leftarrow \mathrm{Concat}(T^{(1)}, T^{(2)}, \ldots, T^{(N_b)})$ ▷ Single sequence of length $L_t = N_b \cdot L_c$

// 2. Build 2D grid over local text tokens. Initialize $G_t \in \mathbb{N}^{B \times 3}$ as follows:
$G_t[:, 0] \leftarrow N_b$ ▷ first grid dimension = chunk index
$G_t[:, 1] \leftarrow L_c$ ▷ second grid dimension = intra-chunk position
$G_t[:, 2] \leftarrow 1$ ▷ dummy spatial axis to share ApplyRoPE API for both video and text tokens

// 3. Compute query, key, and value representations
$Q \leftarrow \mathrm{ProjectAndNormalize\_Video}(X, W_q)$ ▷ Video queries
$K \leftarrow \mathrm{ProjectAndNormalize\_Text}(T, W_k)$ ▷ Local text keys
$V \leftarrow \mathrm{Project\_Text}(T, W_v)$ ▷ Local text values

// 4. Apply RoPE using video and text grids
$\widetilde{Q} \leftarrow \mathrm{ApplyRoPE}(Q, G_v, \Omega)$ ▷ Encode video tokens with (frame, height, width) positions
$\widetilde{K} \leftarrow \mathrm{ApplyRoPE}(K, G_t, \Omega)$ ▷ Encode text tokens with (chunk, intra-chunk) positions

// 5. Multi-head cross-attention over all concatenated chunks
$\hat{X} \leftarrow \mathrm{MultiHeadAttention}(\widetilde{Q}, \widetilde{K}, V)$ ▷ Attend from each video token to all local text tokens across chunks
return $\hat{X}$ ▷ Video features updated with chunk-aware local text information

6.3. Related work

The proposed work is in line with recent studies that address the limitations of cross-attention mechanisms within generative Text-to-Video (T2V) frameworks based on Diffusion Transformers (DiTs). A closely related concept is "Segmented Cross-Attention" introduced in Presto [57], where a prompt is divided into sub-captions using a LLM, each aligned to a specific temporal segment of the video. This method is a parameter-free mechanism for generating long-range videos that follow a sequence of narrative instructions derived from the main caption. Similarly, DiTCtrl [10] is a training-free method that enables multi-prompt video generation by controlling attention to create smooth transitions between different textual conditions over time. These methods aim to improve narrative coherence using explicit sub-prompts that are generated purely from text or are explicitly provided, while still relying on modulation of the global attention pathway. Although PhysVid also aligns textual information with local temporal segments, its objective and mechanism are distinct. In contrast to these methods, our method does not redesign or modulate the core attention module. Instead, we introduce new, separate cross-attention blocks as a modular addition to a pretrained model, specifically to integrate the chunk-wise generated physics prompts, thereby complementing the global prompt without affecting its attention pathways.

6.4. Limitations and future scope

Although video-understanding capabilities in VLMs have improved significantly in recent years, they are still prone to hallucination and can produce information that is completely incorrect or misaligned with the visual content presented. This fundamental challenge currently limits their ability to reliably extract physics information from longer videos or videos with complex spatiotemporal physical content. Furthermore, annotating larger datasets with VLM also requires an additional compute budget. Another challenge is scalability to larger models, since the model size can increase quickly due to additional layers in each transformer block. Therefore, an observed improvement in the physical awareness of the resulting model comes at the cost of slower training and inference over the corresponding baseline. However, this challenge can be mitigated to some extent with the help of advanced techniques for faster sampling, such as model distillation. A more theoretical limitation is the classic train-test distribution mismatch, since, during inference, VLM does not have visual input to generate local annotations and must rely on global text alone. However, the video generator always sees the same interface, which is a sequence of local physics-aware text prompts. The mismatch therefore lies only in the upstream prompt generation bounded by the consistency with which the VLM maps global descriptions to local physics statements with and without visual input. Our experiments on two benchmarks indicate that this does not prevent robust gains in the Physical Commonsense (PC) score. Finally, as described in Sec. 3, while including global T2V prompt in the instruction to the VLM during the annotation of a video chunk helps generate annotations that are aligned with the global prompt, they do not explicitly prevent semantic misalignment of annotations among different video chunks. Future work could explore measures to reduce the computational cost of additional local pathways and improve alignment of locally extracted physics information across all chunks.

7. VLM Instructions

In this section, we provide the details on the instructions given to the VLM for different use cases.

7.1. Physics grounded video chunk annotation

Figure 8 shows the VLM input instruction used to generate physics grounded annotations for video chunks prior to training. During annotation, the global T2V caption is appended to the VLM instruction along with a contiguous chunk of frames from the input video, as discussed in Sec. 3.

7.2. Counterfactual annotation

Figure 9 shows the VLM input instruction used to generate the counterfactual prompt based on incorrect physics. The counterfactual annotation in our method relies only on the generated "positive" local prompt for a given chunk and does not use any other information. As discussed in Sec. 3, this helps prevent the generation of physically correct descriptions which are undesirable in this phase.

7.3. Physics-grounded local prompt generation during inference

During inference, the visual data is not available. However, we still need the local physics based instructions to be provided as input to the model. To ensure this, we use the VLM instruction shown in Fig. 10. This instruction relies only on the information contained in the global T2V caption to generate a coherent set of physically correct local prompts.

8. Annotation Examples

In Fig. 11, we visualize the global and local annotations generated by VLM for an example in the training data set, together with the representative frames for each chunk. Similarly, during inference, Fig. 12 we provide the local annotations generated by the VLM for an example caption, along with the representative frames from the video chunks generated using these annotations.

You will be provided a short video clip taken from a longer video. In addition, you will also receive a caption as input. The caption describes the overall event or scene happening in the longer video and may contain information that is not visible in the short clip. The duration of the clip is less than one second. Your task is to provide a structured description of the physical phenomena grounded in the clip, focusing only on VISIBLE elements in the clip and not on any elements that are not visible. Your description would be used to recreate the short video clip in a physically accurate manner by a downstream video generator, therefore it needs to be physically accurate and consistent with the visible elements in the clip. The information contained within the description should only be enough to describe the physical phenomena contained within that small time segment (less than a second). The description should not contain contradictory statements about events observed in the clip.

Perform the following reasoning steps:
1. Understand the given video with the help of the accompanying caption, focusing on the events happening in sequential order.
2. Analyze the visible elements in the video, including objects, people, animals, and environmental features.
3. Analyze relevant physics observations related to VISIBLE elements and how they OBEY physical laws, considering the following domains:
  - a. Dynamics (motion, forces, energy, momentum): understand what is moving, how it is moving and why is it moving
  - b. Shape (deformation, elasticity): understand the shapes of visible objects and if they are deforming or maintaining their shape
  - c. Optics (illumination, ambience, reflections, refractions, shadows): understand the lighting conditions, reflections, and shadows
4. Based on step 3, think about how these physics principles could be structured as a prompt to a video generator so that it can recreate the video.
5. Structure your response as a JSON string according to the example below. Only include observations that are clearly visible in the video. You will be REWARDED for generating statements that are GROUNDED in physics and VERIFIABLE from the video, and PENALIZED for generating statements that are incorrect in physics, not verifiable from the video, not relevant to any physical phenomena, or copying statements from prompt. Maximize rewards and minimize penalties.

Follow comments in the example below to guide your reasoning:
```
{
 "visible_elements": ["sports car", wheels, "road", "trees", "sunlight", "shadows", "reflections"],
 "thinking": "", //Think about what are the most important physical properties that would help a downstream video generator recreate the exact same video. Cannot be blank.
 "physics": "The car's speed is consistent with its motion. The road texture moves backward as the car moves forward. The rotation speed of the wheels matches the car's speed on the ground. The car's shape remains consistent as it moves. The wheels maintain their circular shape while rotating. The lighting is consistent with a sunny day. Trees cast shadows on the ground according to the position of the sun. Reflections on the car's surface change as it moves.", //Explain briefly how the video obeys physics laws. Cannot be blank.
}
```

Now, let's analyze the following video clip along with its caption given below. Proceed step-by-step as instructed above.
Video caption:

Figure 8. VLM instruction to generate the physics caption for a video chunk

9. Additional Results

9.1. Qualitative examples

To supplement the examples in Fig. 1, we visualize additional results generated by our method in Fig. 13 with comparisons to Wan-14B. In Fig. 14, we also provide additional qualitative results from the ablation study discussed in Sec. 4.3.3, to supplement Fig. 7. These examples are also available as videos along with additional video examples on our project website.

9.2. Similarity metrics

We evaluated PhysVid on *four* similarity metrics as shown in Tab. 4. As can be observed, the results remain consistent, showing a slightly increased FVD score relative to the finetuned baseline, which corroborates the previously noted minor compromise in content fidelity (see Tab. 1) in exchange for substantial improvements in physical realism.

Model	LPIPS↓	FVD↓	SSIM↑	PSNR↑
Wan 1.3B	0.703	417.352	0.217	8.625
Wan (finetuned)	0.671	302.465	0.239	9.379
PhysVid	0.679	318.087	0.240	9.234

Table 4. Additional metrics based on similarity (2048 sample pairs)

10. Other details

Configuration. Table 5 lists hyperparameter configurations and other relevant settings for training and inference.
VLM overhead. The average VLM overhead for all prompts generated per sample is $\approx 17.22 \pm 0.99$ seconds. Including the denoising loop runtime quantified in Tab. 5, overall, PhysVid approach takes 110 seconds per video, which is near third ($0.35\times$) of 310 seconds per video latency of Wan-14B. All inference run times in our work are reported with *bfloat16* precision on a single B200 GPU with a batch size of 1.

You will be provided a physics-rich description of a scene. Perform the following reasoning steps:
1. Identify key elements, objects and environmental features identifiable from the scene description.
2. Each statement in the input belongs to one of three categories of statements: dynamics, shape, and optics. Look at each input statement one-by-one and identify which category it belongs to.
3. As a physicist, predict what would happen instead if physics laws were NOT obeyed within that category.
4. Based on step 3, think about a video that would result from VIOLATION of physics laws in each category and generate a description for that video. Ensure that your description clearly violates the physics laws within the identified category.
5. Structure your response. You will be REWARDED for generating statements that are unrealistic in physics, and PENALIZED for generating statements that are correct in physics, not relevant to any physics phenomena, or copying input. Maximize rewards and minimize penalties.

Example Input:
The car's motion is smooth. The road texture moves backward as the car moves forward. The rotation speed of the wheels matches the car's speed. The car's shape remains consistent as it moves. The wheels maintain their circular shape while rotating. The lighting is consistent with a sunny day. Trees cast shadows on the ground according to the position of the sun. Reflections on the car's surface change as it moves.

Example output:
{
"visible_elements": ["car", "wheels", "road", "trees", "sunlight", "shadows", "reflections"],
"thinking": "To describe a video that violates physics laws, I need to focus on unrealistic motion dynamics, shape deformations, and incorrect optical effects related to the car, its wheels, the road, trees, sunlight, shadows, and reflections.", //Explain briefly how a video that does NOT follow physics laws would look like. Cannot be blank.
"physics": "The car's speed varies unrealistically. The road texture moves forward as the car moves forward. The rotation speed of the wheels does not match the car's speed. The car's shape changes significantly as it moves. The wheels lose their circular shape while rotating. The lighting is inconsistent with a sunny day. Trees do not cast shadows on the ground according to the position of the sun. Reflections on the car's surface remain static as it moves."
}

Explanation of how the output was generated:
What would happen if physics laws were NOT obeyed within each category:
- Dynamics: The car's speed could vary unrealistically. The road texture could move forward as the car moves forward. The rotation speed of the wheels may not match the car's speed. The car could be flying or hovering above the ground.
- Shape: The car's shape could change as it moves. The wheels might not be circular shape while rotating.
- Optics: The lighting would be inconsistent with a sunny day. Trees will not cast shadows on the ground according to the position of the sun or there may be irregularly shaped shadows not matching the object's shape. Reflections on the car's surface may remain static as it moves.

Now, let's consider the input given below STEP-BY-STEP.
Input:

Figure 9. VLM instruction to generate the counterfactual physics caption

You will be provided a caption describing an event or a scene. Your task is to provide a set of SEVEN captions describing the physical phenomena grounded in the original caption. The set of captions would be used to generate a video that is physically accurate and grounded in the original caption. Each caption would be used sequentially by the downstream video generator to generate a short chunk of video. Each caption therefore needs to be physically accurate and consistent with the original caption. Each caption should describe a small time segment of the overall event or scene. The set of captions should together cover the entire event or scene described in the original caption leading to a coherent video when stitched together. The information contained within each caption should only be enough to describe the physical phenomena contained within the corresponding small time segment (less than a second). A caption should not describe an event that is not possible within the small time segment, but it can build on events from previous segments. Do not describe any elements that are not visible in the scene or are impossible to visualize. Do not output any statements that directly contradict the input.

Perform the following reasoning steps:
1. Imagine a short, few seconds scene based on the caption. Identify key elements, objects and environmental features identifiable from the caption that may be visible in the scene. Do not include any elements that are not visible.
2. Analyze relevant physics observations related to these elements and how they OBEY physical laws, considering the following domains:
  - a. Dynamics (motion, forces, energy, momentum): understand what is moving, how it is moving and why is it moving
  - b. Shape (deformation, elasticity): understand the shapes of objects as mentioned in the caption and if they are deforming or maintaining their shape
  - c. Optics (illumination, ambience, reflections, refractions, shadows): understand the lighting conditions, reflections, and shadows
3. Based on step 2, think about how these physics principles could be structured as a set of seven temporally correlated prompt sequence to a video generator so that it can recreate the scene described in the original input caption.
4. Structure your response as a JSON string according to the example below. Output only those statements that are relevant to the input. You will be REWARDED for generating statements that are GROUNDED in physics and the original input, and PENALIZED for generating statements that are incorrect in physics, completely unrelated to the input, not relevant to any physical phenomena, or copying statements from input. Maximize rewards and minimize penalties.

Example Input:
A car moving along the road on a sunny day.

Example Output (follow the comments in the code for instructions on how to generate outputs):
```
  {
    "thinking": "To describe a video that follows physics laws, I need to focus on realistic motion dynamics, shape consistency, and accurate optical effects related to the car, its wheels, the road, trees, sunlight, shadows, and reflections.", //Think about what are the most important physical properties that would help a downstream video generator produce the exact same video. Cannot be blank.
    "visible_elements": ["sports car", wheels, "road", "trees", "sunlight", "shadows", "reflections"],
    "physics": [
            "The car accelerates smoothly along the road, with its speed consistent with its motion.",
            "The road texture moves backward relative to the car's forward motion, creating a realistic sense of movement.",
            "The wheels rotate at a speed that matches the car's forward velocity, ensuring proper traction and motion dynamics.",
            "The car maintains its shape as it moves, with no visible deformations or alterations.",
            "The wheels retain their circular shape while rotating, demonstrating structural integrity.",
            "The lighting conditions reflect a sunny day, with consistent brightness and color throughout the scene.",
            "Trees cast accurate shadows on the ground based on the sun's position, enhancing the realism of the environment."
          ] //Describe SEVEN different temporally correlated captions that help recreate the imaginary video that obeys physical laws. Each caption should describe a small time segment of the overall event or scene described in the original caption leading to a coherent video when stitched together. Each caption should be physically accurate and consistent with the original input and the preceding captions.
  }
```

Now, let's analyze the following caption. Proceed step-by-step as instructed above.
Caption:

Figure 10. VLM instruction to generate the local physics captions for all the chunks to be generated at once during inference

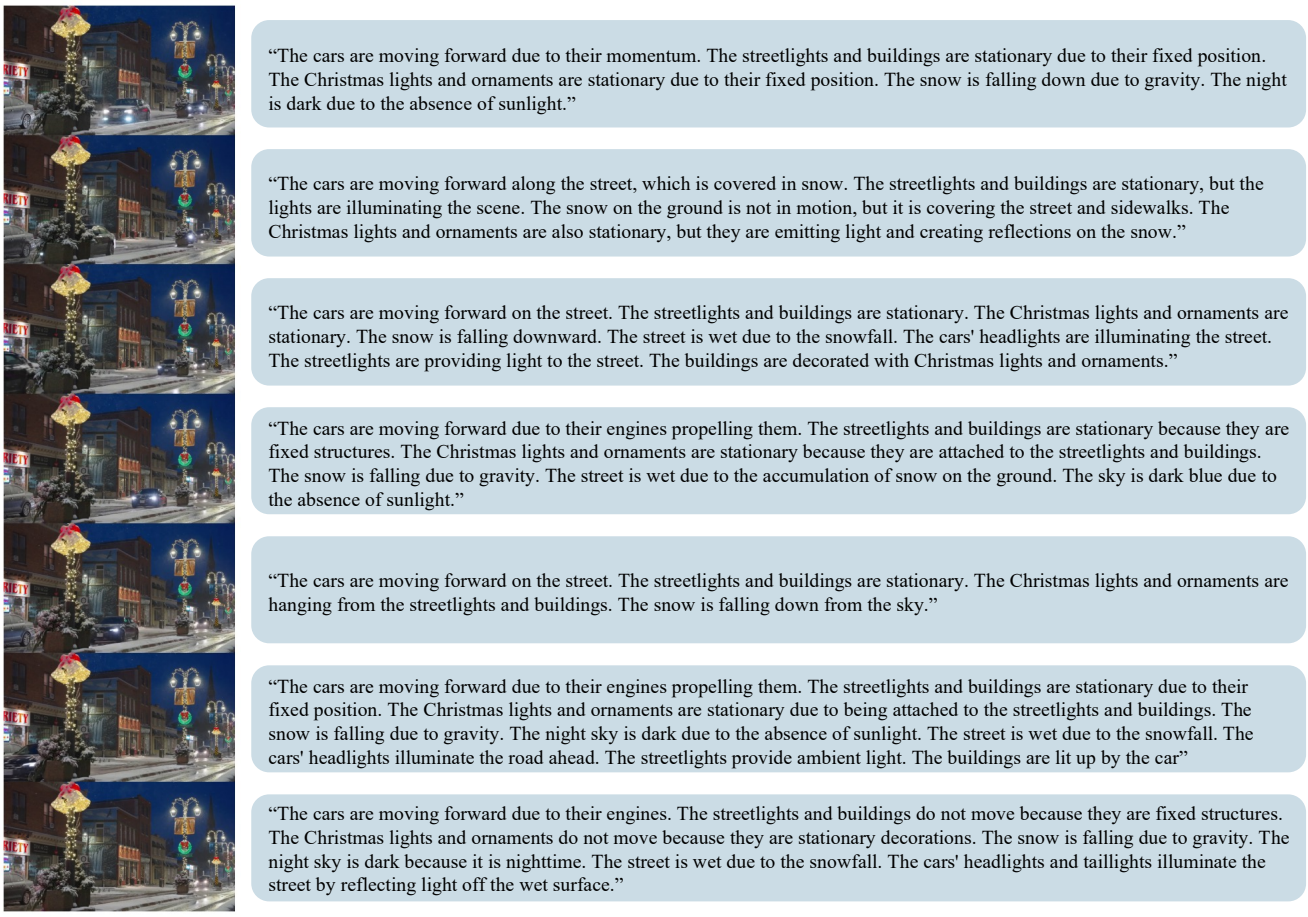

Figure 11. A sample from the data with annotations generated by a Vision Language Model (VLM). The topmost text is the global prompt and the local annotations are listed sequentially alongside a representative frame from each chunk.

“A car driving on a snowy road.”

“The car moves forward on the snowy road, with its tires gripping the slippery surface to maintain traction.”

“The tires do not grip the slippery surface, and the car maintains traction despite the lack of friction. The snowy road is solid and not slippery at all. The car moves forward effortlessly on the snowy road, with its tires floating above the ground.”

“The car's speed is consistent with its motion, adapting to the snowy conditions for safety and efficiency.”

“The car's speed is inconsistent with its motion, not adapting to the snowy conditions for safety and efficiency. The snowy conditions have no effect on the car's motion. The car moves at a constant speed regardless of the snowy conditions.”

“The wheels rotate at a slower pace due to the increased friction from the snow, demonstrating the impact of the road's texture on motion dynamics.”

“The wheels rotate at an unrealistically fast pace due to the increased friction from the snow, demonstrating the impact of the road's texture on motion dynamics. The road's texture moves backward as the car moves forward, defying the expected relationship between the car's motion and the road's texture.”

“The car maintains its shape as it drives, with no visible deformations or alterations due to the cold weather.”

“The car becomes misshapen or deformed due to the cold weather. The car's shape changes due to the cold weather. The car shrinks or expands due to the cold temperature.”

“The snow on the road appears freshly fallen, with no visible signs of melting or disturbance, indicating recent snowfall.”

“The snow remains perfectly undisturbed and does not melt or change shape despite exposure to sunlight and temperature fluctuations.”

“The sky is clear, providing consistent lighting conditions that enhance the visibility of the snowy landscape.”

“The colors of the sky shift rapidly, creating an otherworldly atmosphere. The visibility of the snowy landscape varies as the sky changes its appearance. The sky's lighting conditions are inconsistent, causing the snowy landscape to be illuminated by different light sources at different times.”

“The car casts a shadow on the snow, accurately reflecting the sun's position and time of day, adding to the realism of the scene.”

“The sun's position and time of day are inconsistent with the shadow cast by the car. The car casts an inaccurate shadow that does not reflect the sun's position and time of day.”

Figure 12. A set of physics grounded and physics counterfactual prompts generated during inference. The representative frames from the generated video chunks are shown on the right.

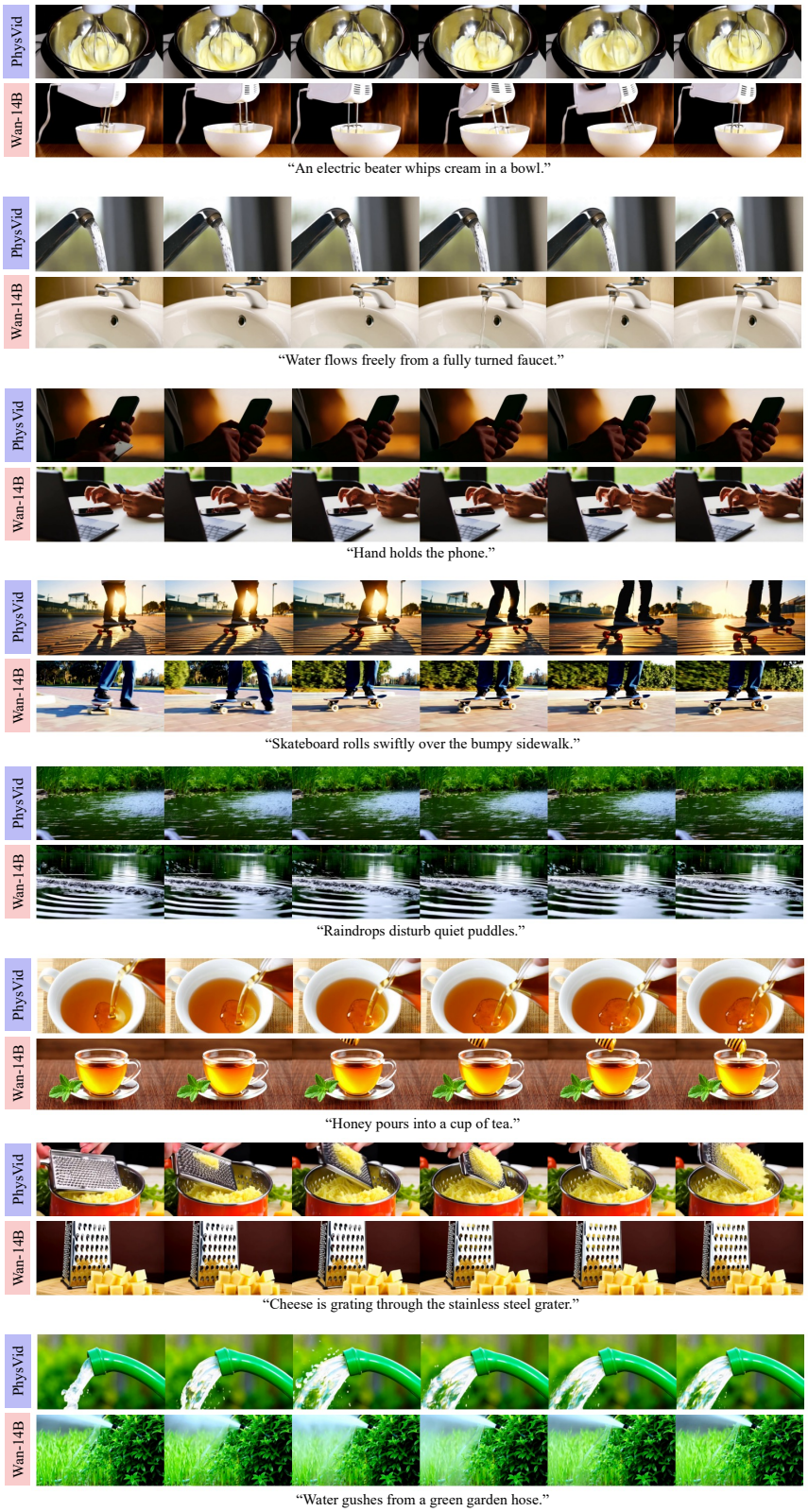

Figure 13. Additional comparisons between *PhysVid* and *Wan-14B*. Captions are from VideoPhy.

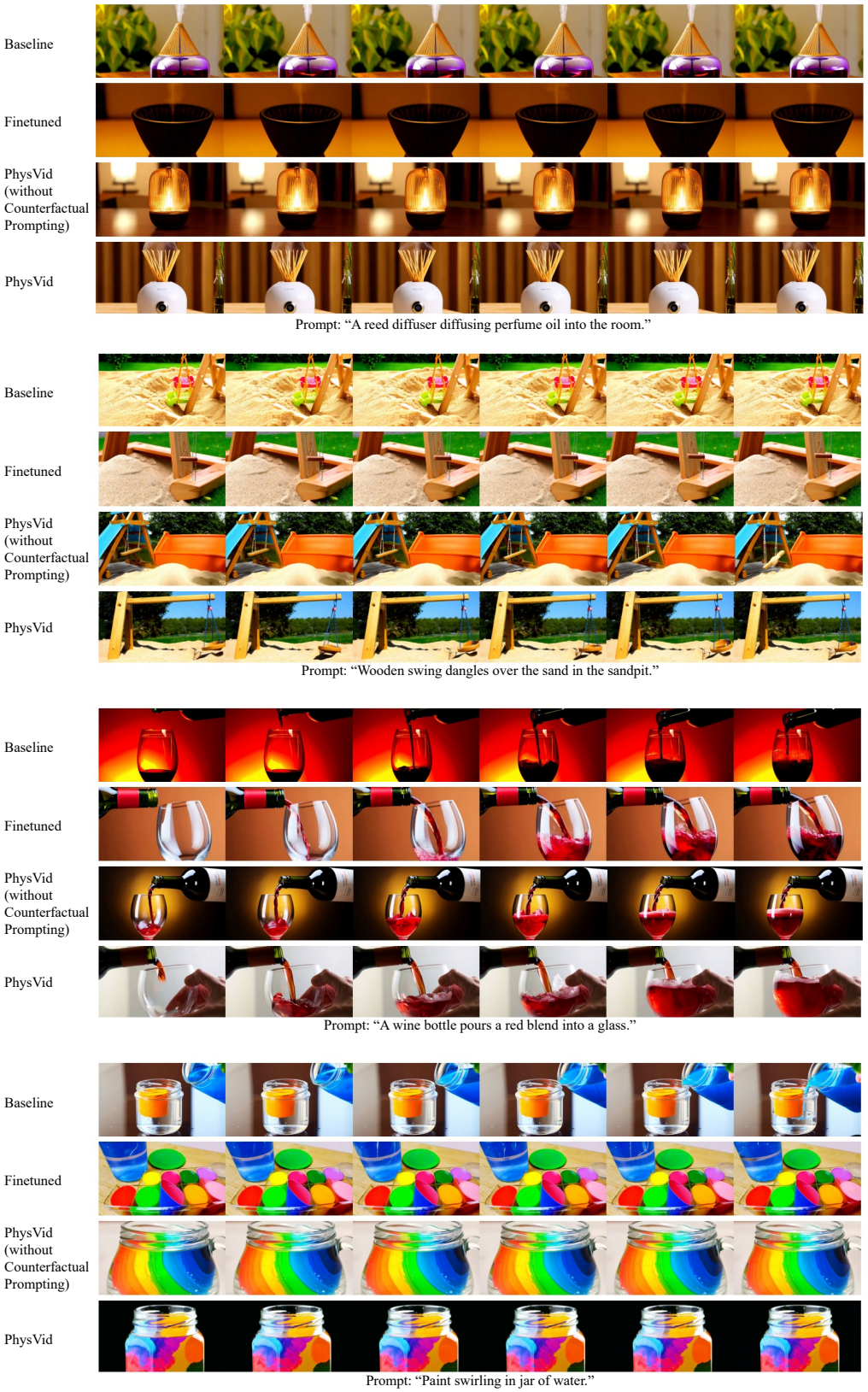

Figure 14. Supplementary qualitative results from the ablation study. All prompts are from VideoPhy.

Table 5. Configurations

Training	
Base architecture	Wan-1.3B
Additional parameters (M)	400
Effective batch size	64
Number of steps	3000
Number of epochs	4
Learning rate (Stage 1: 1000 steps, frozen base layers)	1×10^{-5}
Learning rate (Stage 2: 2000 steps, full architecture)	2×10^{-6}
Loss	Flow Matching
Optimizer	AdamW
Timestep Shift Factor	8
Number of Latent Frames per Chunk	3
Number of Latent Chunks	7
Inference	
Number of Denoising Steps	50
Guidance Scale	6
Wan-1.3B Latency per Video (s)	66
Wan-14B Latency per Video (s)	310
PhysVid Latency per Video (s)	93
Video	
Resolution	832×480
FPS	16
Duration (s)	5.06
Frames	81