What Happens Next? Next Scene Prediction with a Unified Video Model

Supplementary Material

10. Causal Consistency Reward

Here, we present the prompt for causal consistency reward. Given a preceding scene description and a generated video caption, the prompt requires the judge model to evaluate whether the two are logically consistent. In addition, it requires verifying that the next scene does not excessively repeat the content of the preceding one.

Template 10.1: Causal Consistency Reward

```
 You will be given two text descriptions:
(1) A preceding scene (context)
(2) A generated video description
(caption)
Your task is to determine whether there
is logical causal consistency between
the scene and caption.  Additionally,
evaluate whether the caption is
semantically redundant with the scene
description---that is, whether it repeats
most of the same people, actions, objects,
or setting, rather than continuing with
new content.
Criteria:
Pass if:
-- The caption shows logical causal
continuity from the scene description.
-- The caption describes a clearly
new moment with different actions,
progression, or consequences.
-- The overlap in content (people,
actions, objects, setting) is low.
Fail if:
-- The caption is disconnected or
implausible in progression.
-- The caption largely repeats the same
content.
-- The two descriptions refer to the same
moment or situation.
Input:
Scene Description:  {scene_description}
Caption:  {caption}
Output:
Verdict:  Pass / Fail
```

11. NSP Dataset Curation

11.1. Prompts

Here, we present the prompts for generating preceding scene descriptions and for verification.

Preceding Description Prompt Template

```
Task:  Predict the most likely prior
event or activity using commonsense
knowledge based on the observed video.
The video shows:  [Current visual
scene or activity]
Description:  [What likely happened
just before this moment]
= Guidelines:  Use real-world causal
reasoning.  Describe an earlier event
that leads to the current scene,
without repeating existing actions
or characters.
Example:
The video shows:  a boy is tying his
shoelaces.
Description:  He was in a hurry to
catch the school bus outside.
```

Verification Prompt Template

```
You will be given two text
descriptions:
1.  A preceding scene (context)
2.  A generated video description
(caption)
Your task:  Determine whether the
caption is semantically redundant
with the scene description or if it
provides new, causally consistent
content.
Label as ``Pass'' if the caption
describes a new moment with causal
continuity; otherwise label as
``Fail''.
```

11.2. Data Filtering

To ensure high-quality supervision, we apply the following filtering rules: (1) retain video clips shorter than 5 seconds at 16 fps (maximum 81 frames); for longer clips, only the first 5 seconds are used; (2) keep videos with a resolution of at least 832×480 pixels; (3) resize all training videos to 832×480 during training. Each image in the image datasets is treated as a single-frame video with a resolution of 512×512. While the current setup adopts fixed spatial resolutions, future extensions will explore adaptive resolution training.

Table 5. Summary of datasets used in different training stages for text-to-video pre-training.

Dataset	Quantity	Stages
BLIP-3o (Image)	27M	20M (Stage 1), 7M (Stage 2)
VidGen	1.0M	Stage 2
OpenVid	0.31M	Stage 2
OpenS2V	1.33M	Stage 2
OpenHumanVid	10.8M	Stage 3

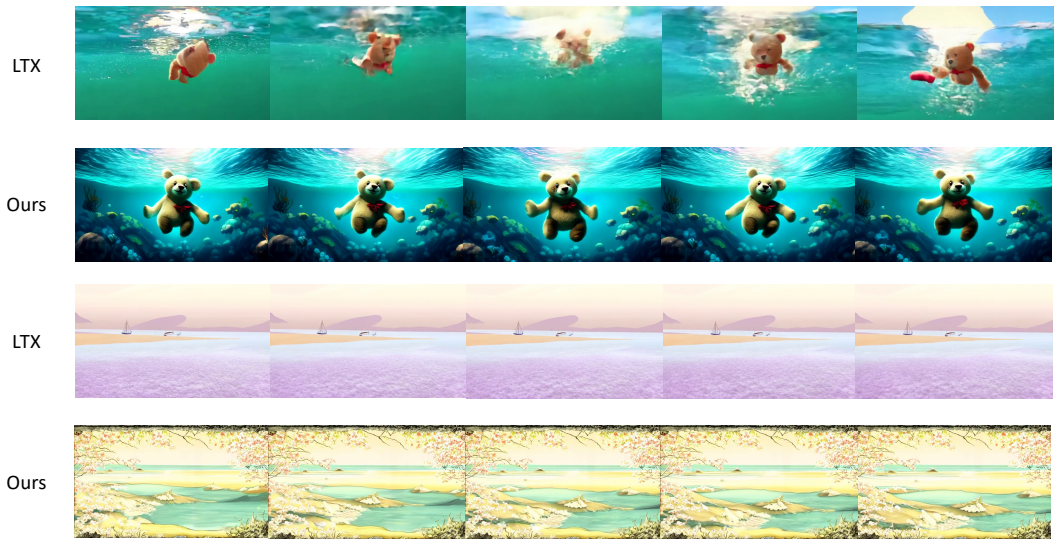

Figure 7. Qualitative comparisons between the original LTX model and our text-to-video pre-trained model. Prompt 1: A teddy bear is swimming in the ocean.; Prompt 2: A beautiful coastal beach in spring, waves lapping on sand by Hokusai, in the style of Ukiyo. Video results are attached in the supplementary material.

12. Text-to-video Pre-training

12.1. Dataset Statistics

Detailed dataset statistics for the different training stages for text-to-video pre-training are presented in Table 5.

12.2. Results

As shown in Fig. 7, our text-to-video pre-trained model generates videos with enhanced visual fidelity and better semantic alignment, demonstrating the advantages of replacing the original text encoder and adopting our unified architecture.

We also provide detailed comparisons on the VBench [19] benchmark in Table 7.

12.3. Effect of Removing Pre-training

We conduct an ablation experiment where the model is trained directly on the NSP dataset without text-to-video pre-training. As shown in Table 6, removing pre-training leads to a drastic performance drop, yielding a causal consistency score of only 0.06. Since our base generator is LTX, a text-to-video model, text-to-video pre-training is essential for enabling effective transfer of semantic information from the Qwen-VL understanding module to the LTX generator.

Table 6. Causal consistency comparison between models with and without text-to-video pre-training on the NSP test dataset.

Training Stage	Causal Consistency
Pre-training	0.54
SFT	0.60
RL	0.73
w/o PT	0.06

13. Implementation Details

We build our framework upon the Qwen-VL 2.5 (7B) model [1] and the LTX Text-to-Video (0.9.0) model [15]. Our architecture employs 256 learnable queries and a maximum sequence length of 1024 tokens. For classifier-free

Table 7. Detailed Comparisons on VBench.

Dimension	LTX	Stage 1	Stage 2	Stage 3
Quality Dimensions				
Subject Consistency	0.898	0.902	0.960	0.961
Background Consistency	0.944	0.957	0.968	0.970
Temporal Flickering	0.986	0.986	0.991	0.993
Motion Smoothness	0.974	0.987	0.993	0.993
Dynamic Degree	0.625	0.675	0.300	0.303
Aesthetic Quality	0.544	0.566	0.624	0.628
Imaging Quality	0.529	0.427	0.570	0.573
Quality Score	**0.780**	**0.782**	**0.802**	**0.805**
Semantic Dimensions				
Object Class	0.749	0.725	0.839	0.798
Multiple Objects	0.364	0.152	0.370	0.412
Human Action	0.866	0.692	0.856	0.872
Color	0.816	0.716	0.836	0.850
Spatial Relationship	0.572	0.233	0.517	0.523
Scene	0.526	0.424	0.485	0.510
Appearance Style	0.220	0.214	0.245	0.241
Temporal Style	0.223	0.207	0.232	0.235
Overall Consistency	0.246	0.216	0.245	0.249
Semantic Score	**0.674**	**0.549**	**0.686**	**0.695**
Total Score	**0.759**	**0.736**	**0.779**	**0.783**

guidance (CFG) [17], 10% of conditions are randomly dropped during training, and a CFG scale of 3.0 is applied during inference to balance generation quality and diversity. During inference, to ensure fair comparison with baselines, we generate 65 frames at a resolution of 832×480 without applying negative guidance. The sampling step is set to 50. Within the connector module, the RMSNorm layer weight is initialized to $\sqrt{5.5}$ and equipped with a learnable scale factor, initially set to 0.01. This design enables dynamic scaling of the connector's output, which is essential for maintaining training stability and enhancing overall performance.

For all the aforementioned prompts involved in reward computation, metric evaluation, and NSP dataset curation, we utilize the Claude 3.7 Sonnet model as the judge model to ensure consistent reasoning quality and alignment.

Detailed implementations for different stages are explained as follows.

Pre-training. For stage 1, we train the model for 3 epochs with a batch size of 32. For stage 2, we train for 5 epochs with a batch size of 8, and for stage 3, 2 epochs with a batch size of 8. The model is optimized using the Prodigy optimizer [32] with an initial learning rate of 1.0. All stages are trained on 32 80G A100 GPUs.

Supervised Fine-tuning. For the NSP task, we train the model on 32 80G A100 GPUs with a batch size of 8, using the Prodigy optimizer [32] with an initial learning rate of 1.0.

Reinforcement Learning. For reinforcement learning, the model is trained on 16 80G A100 GPUs with a gradient accumulation step of 8 for a total of 60 optimization steps. We use a batch size of 1, a CFG scale of 3.0, and an input resolution of $480 \times 832 \times 65$. The sampling step is set to 20. For each input, the model generates 24 video candidates using identical noise seeds. Among these, we adopt a Best-of-N sampling strategy ($N = 8$), where the top 4 and bottom 4 samples ranked by reward scores are selected for reward optimization. This approach encourages the model to learn from both high- and low-quality generations, improving stability and reward sensitivity during optimization.

14. Comparison with Related Tasks

Compared to *next shot generation* [16], which predicts the immediate camera shot within the same scene focusing on short-term continuity, and *multi-scene generation* [13], which emphasizes visual consistency across different scenes, our task targets long-term causal and temporal reasoning across consecutive scenes.

Table 8. Efficiency comparison with previous methods.

Methods	GPU Memory Usage	Time
LTX	14.9G	12s
Wan	16.9G	2m22s
Omni-Video	37.6G	2m16s
Ours	26.4G	13s

15. Efficiency Comparison

Here, we compare the memory usage and sampling time of our model with several prior video generation models, including the original LTX model [15], the Wan 2.1 1.3B model [47], and the open-source unified video model Omni-Video [43]. The results are summarized in Table 8. Compared with Wan and Omni-Video, our model achieves nearly a 10× speed-up in sampling, owing to its efficient generation architecture and connector design. Relative to Omni-Video, our model also requires less GPU memory. All comparisons are conducted using an input resolution of $480 \times 832 \times 65$, 50 sampling steps, on a single 80G A100 GPU.