VideoWorld 2: Learning Transferable Knowledge from Real-world Videos

Supplementary Material

In this supplementary material, we provide additional details and analyses:

- Sec. A describes the training setup, the structure of the dLDM, and other implementation details.
- Sec. B presents extended comparisons with related methods and additional experiments.
- Sec. C provides further analysis of Video-CraftBench.
- Sec. D provides additional visualizations of both our method and the baselines.

A. Implementation Details

Training details. We present the detailed training configurations of dynamics-enhanced latent dynamics model and auto-regressive transformer in Tab. A.1. To improve training efficiency, dLDM first applies a short warm-up where the latent codes are optimized solely using the original reconstruction objective. This warm-up is similar to the training strategy of the original VideoWorld LDM, enabling the latent codes to rapidly learn to compress visual changes and motion dynamics, while allowing the decoder to reconstruct low-fidelity video clips containing agent motion trajectories based on the initial frame and codes. Consequently, when we switch to the disentangled scheme, the decoder can provide robust motion conditioning to stabilize training. The trainable parameters in our framework include the AR transformer, the dLDM autoencoder, the DiT, and the projection layer mapping the latent codes to the DiT.

Dynamics-enhanced latent dynamics model (dLDM). As shown in Fig. 4, the proposed dLDM consists of four components. (i) A causal encoder extracts visual features. (ii) A set of learnable queries attends to the encoder features to extract visual changes and produce the latent dynamic codes. (iii) A decoder reconstructs frames from the latent dynamic codes and the feature of the first frame, providing coarse motion cues. (iv) A pretrained Video Diffusion Model (VDM) generates high-fidelity future frames by taking three inputs: the first frame, the low-resolution decoder outputs, and the latent dynamic codes. The latent codes are injected into the VDM via causal cross-attention, allowing the codes to concentrate on task-relevant dynamics while the VDM handles visual appearance modeling. We provide PyTorch-style pseudocode for the full dLDM and its components in Alg. 1.

Latent pre-training on CALVIN. In Sec. 5.4, we present a latent pre-training experiment where latent dynamic tokens serve as prediction targets. This enables the model to learn manipulation knowledge from unlabeled data, thereby facilitating better adaptation to ground-truth actions. Following LAPA [70]'s protocol, given a task trajectory $x_{0:T}$, we train the AR Transformer to predict the quantized latent dynamic embeddings $\{z_k^n\}_{k=1,n=1}^{K,N}$ (defined in Sec. 3 line 198), conditioned on the language instruction and the first image x_0. We extend the AR transformer's vocabulary to

Config	dLDM	AR Transformer
optimizer	AdamW	AdamW
base learning rate	1e-4	3e-4
weight decay	0.1	0.05
optimizer monmentum	β_1, β_2=0.9,0.99	β_1, β_2=0.9,0.98
batch size	128	256
learning rate schedule	WarmipDecayLR	WarmipDecayLR
warmup iterations	2e+3	5e+3
max iterations	1e+5	5e+4
augmentations	None	None
Training Loss	L2 & Denoising loss	CE loss
Training Target	Recon. & Denoising	Next token pred.

Table A.1. **Training configurations** for the dLDM and auto-regressive (AR) transformer.

Proj Layer	Interplay Attn type	Paper	Block
MLP	cross	52.0	61.3
+self	cross	52.3	61.8
MLP	causal cross	69.8	78.6
+self	causal cross	**72.5**	**80.9**

Table A.2. LDM/VDM interplay.

Training Strategy	Video-CraftBench Paper	Block
baseline	0.0	28.5
random	0.0	0.0
freeze	31.7	40.2
lora	50.9	62.3
full	68.8	77.5

Table A.3. Training of VDM.

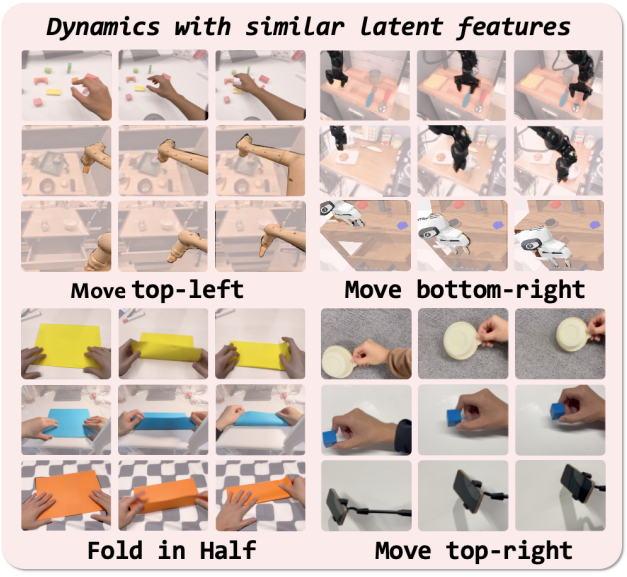

Figure A.1. **Video clips with similar latent dynamic features.** The text below represents the dynamic type.

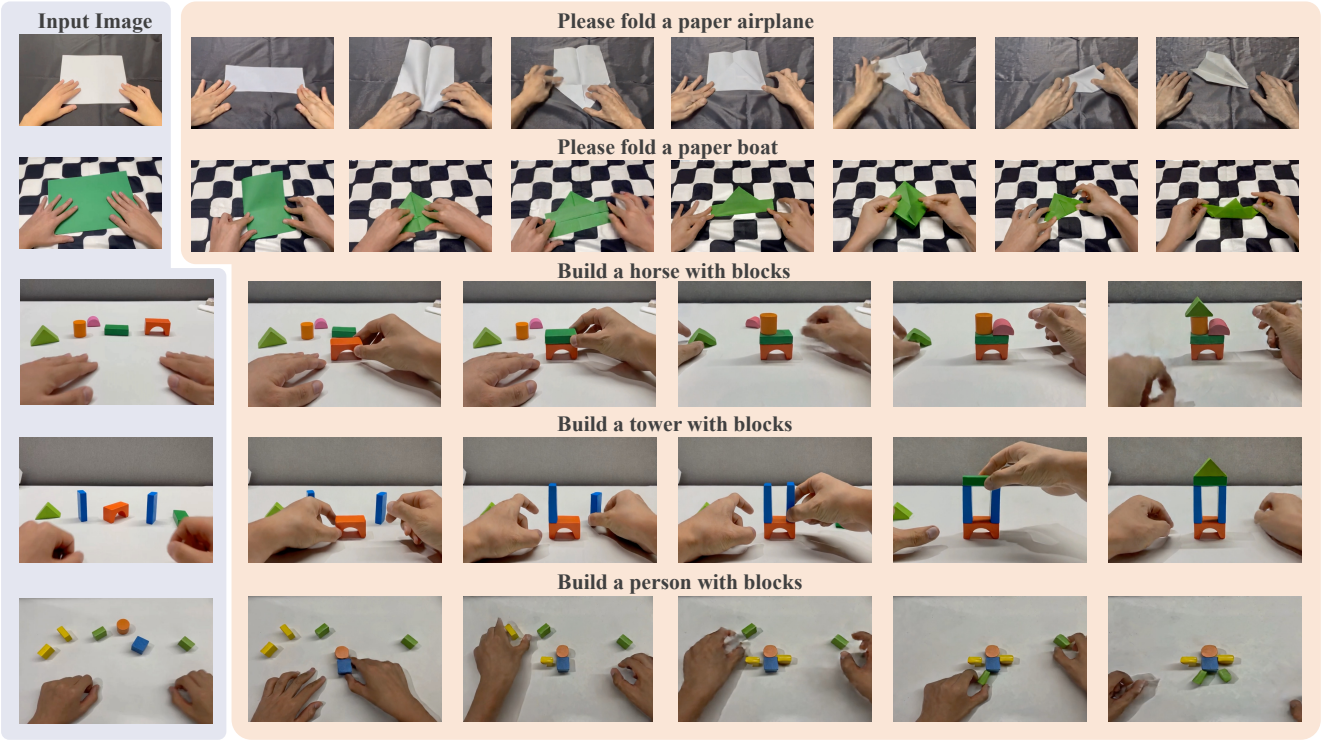

Figure A.2. **Qualitative Results.** VideoWorld 2 learns transferable knowledge and generates long-horizon videos in unseen environments.

include these latent dynamic tokens. Since this pre-training does not rely on ground-truth actions, it can use *any unlabeled video data.*

Following latent pre-training, we further fine-tune the model with ground-truth labels to derive actions executable in the simulation environment. We append a new action head (an MLP layer) to the transformer to map the hidden states, which are originally used for latent token prediction, to real action labels via an ℓ_2 loss. To ensure compatibility with cross-environment data, we exclusively use CALVIN's static camera images during both pre-training and fine-tuning, omitting wrist camera images and state parameters. This latent pre-training should provide the AR transformer with a manipulation prior, leading to better fine-tuning performance.

B. More Comparisons and Ablations

To further clarify the connection and distinctions between VideoWorld 2 and related approaches, we provide extended comparisons by examining methods that explore "disentanglement" and VideoWorld.

Difference with "Disentanglement" in other models. While some video generation models also explore "disentanglement", VideoWorld 2 differs substantially in both objective and methodology. First, in prior works [9, 38, 39, 44, 53, 61, 66], disentanglement typically refers to separating motion from appearance for applications like style transfer or visual editing (e.g., camera movement or object-specific changes). In contrast, VideoWorld 2 targets a more demanding objective: reducing task-irrelevant information to learn transferable visual dynamics for complex long-horizon tasks. Second, technically, many works [9, 38, 39] rely on explicit geometric supervision to isolate motion signals. Others [53, 61] capture only coarse, global motion semantics rather than temporal dynamics, while methods such as [44, 66] depend on handcrafted residual encoding or manually separated parameter groups, and works like [26, 78] focus on pixel-level reconstruction. In contrast, VideoWorld 2 employs a dynamics-enhanced latent dynamics model that suppresses appearance variation and captures task-relevant visual dynamics directly from unlabeled videos, yielding representations suitable for complex task execution, which is beyond the capability of priors.

Difference with VideoWorld. Regarding VideoWorld [51], it struggles to decouple task-relevant dynamics from visual appearance, leading to appearance drift and motion errors in unseen environments (see Fig. B.7). By delegating appearance modeling to the pre-trained VDM, VideoWorld 2 allows its latent space to focus on concise and transferable dynamics rather than appearance details, substantially improving robustness and transferability.

Interplay between VDM and LDM. Tab. A.2 ablates the interplay mechanism between the LDM and VDM, which consists of a projection layer (MLP and causal self-

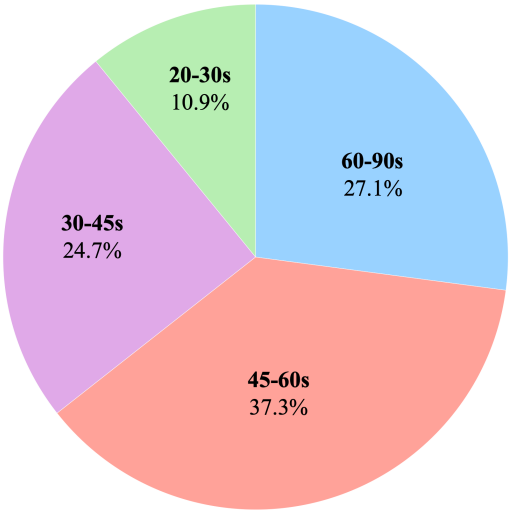

Figure A.3. **Video duration distribution** of Video-CraftBench.

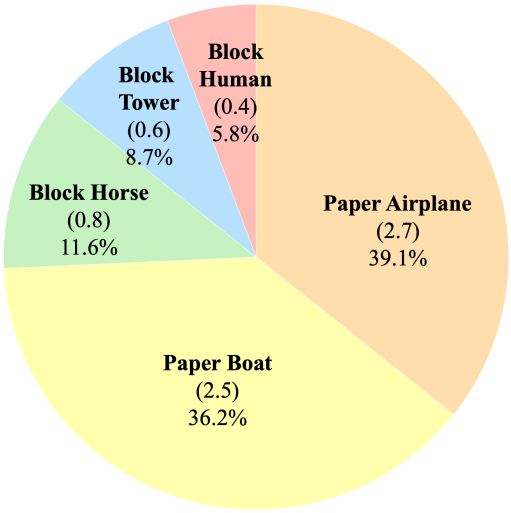

Figure A.4. **Task type distribution** of Video-CraftBench.

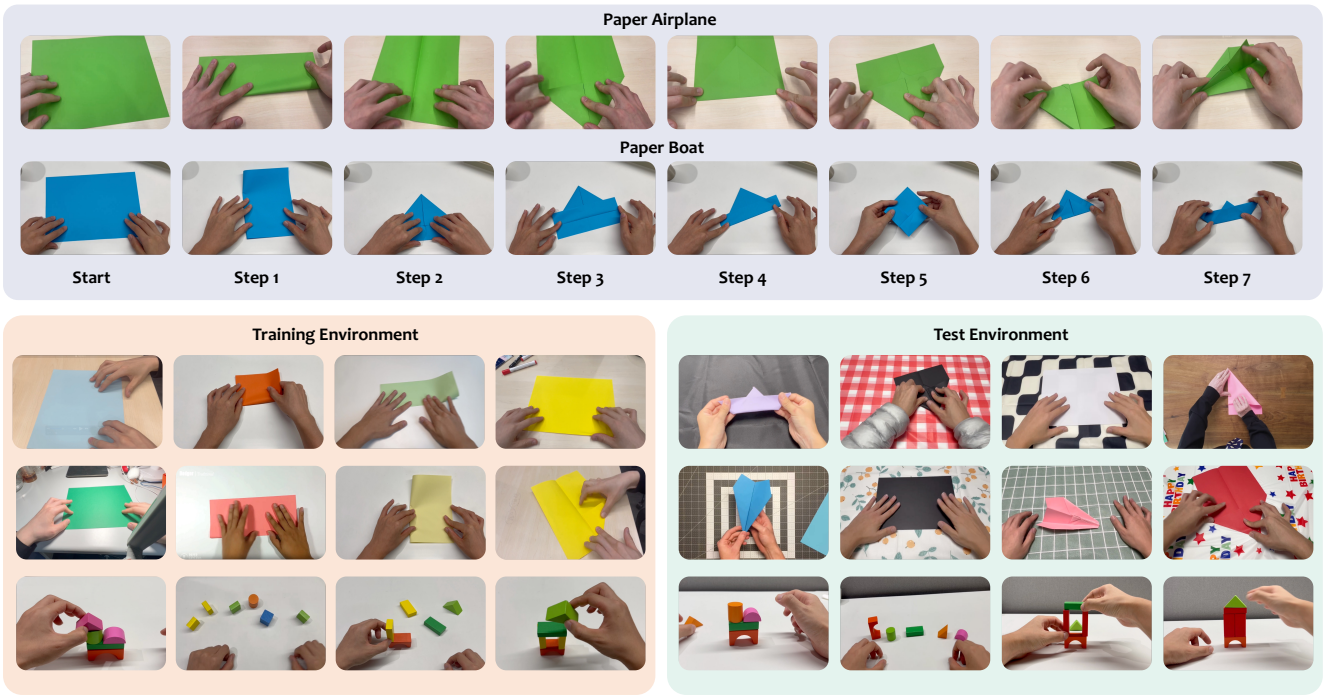

Figure A.5. **Overview of Video-CraftBench** and keys steps of paper folding tasks. Best viewed in color.

attention) and causal cross-attention. The causal cross-attention ensures generation relies solely on the current time step's latents, effectively preventing information leakage. Results show that incorporating self-attention into the projection layer and utilizing causal cross-attention significantly enhance LDM-VDM interaction, validating the effectiveness of our design

Training strategies of VDM. In Tab. A.3, we examine different training configurations for the video diffusion model. Our default approach fully fine-tunes the pre-trained VDM to exploit its appearance priors. In contrast, training a randomly initialized VDM (row 1) leads to model collapse and fails to generate valid videos. We also experiment with freezing the pre-trained VDM and updating only the VQ-VAE components (encoder, quantizer, and projection layer). Although this setup yields performance gains, it falls significantly short of LoRA or full fine-tuning. We believe this gap exists because the VDM needs further adaptation

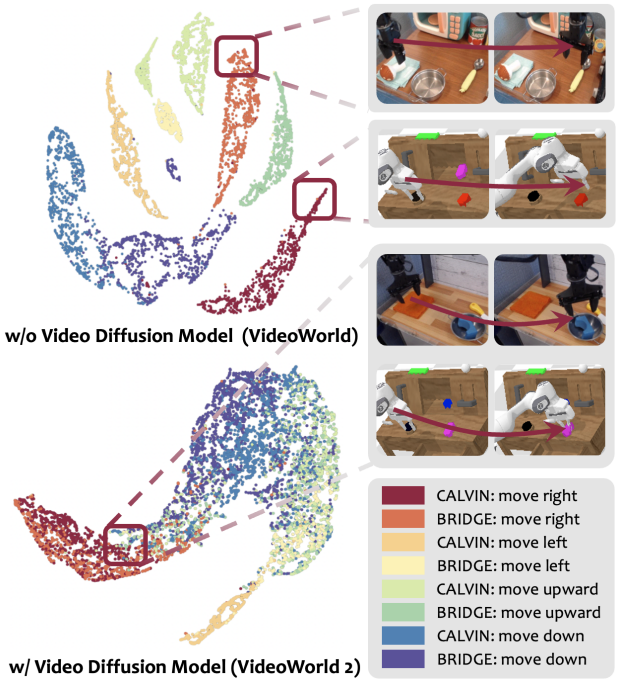

Figure A.6. **UMAP of latent codes.** With a pretrained VDM, latent codes for the same action align more tightly than those in VideoWorld [51] across environments, indicating more consistent and transferable dynamics. (bottom) In distinct environments like Bridge [58] and CALVIN [45], latent representations of the same action (e.g., the robotic arm moving right) are highly similar. (top) Conversely, in VideoWorld, latent codes for the same action exhibit significant divergence across environments and fail to cluster effectively in the UMAP visualization.

to capture the fine-grained manipulation details specific to Video-CraftBench.

C. Details in Video-CraftBench

Key steps for evaluation. In Sec. 4.2, we define seven key steps in the paper folding task to evaluate the task success rate. Fig. A.5 (top) provides a schematic diagram of these key steps.

Train and test environments. Fig. A.5 (bottom) illustrates the Video-CraftBench training and testing environments. To evaluate the model's generalization to new environments, the test set features diverse variations compared to the training set. Specifically, the paper folding test set differs in background, texture, paper appearance, and camera viewpoints, while the block-building varies in initial block arrangement, color combinations, and camera angles.

Task analysis. Fig. A.3 shows the task duration distribution of Video-CraftBench. The largest portion (37.3%) falls within 45–60 seconds, followed by 27.1% at 60–90 seconds, while short tasks (20–30s) make up only 10.9%. This distribution highlights our emphasis on long-horizon tasks.

Algorithm 1: Pseudo codes of dLDM.

```
# Inputs: video:The first frame and its subsequent H frames [1+T, h, w, 3];
# Variables: ldm_q: Learnable embeddings [N,C];
# Functions: CrossAttention();MLP(); up_scale(); down_scale(); FSQ(); Causal3DCNN();pre-trained video diffusion model VDM()
def encoder(video):
    # video:video sequences.[1+T,h,w,3]
    # The encoder consists of a set of encoder layers, composed of Causal3DCNN and down_scale layers.
    f = Causal3DCNN(video);# f:[1+T,h,w,C]
    for layer in encoder_layers:
        # process and spatialtemporally downsample features using Causal3DCNN and down_scale.
        f = layer(f)
    # f:[K, h',w',C]. K=1+T/4
    # Capture dynamic changes in video.
    z = ldm_qformer(f)
    return z, f[:0];# f:[1,h',w',C]. z:[K-1,C]
def ldm_qformer(f):
    # f:features of each frame.[K, h', w', C]
    q_list = []
    for k in range(2, K+1):
        f_k = f[:k];# f_k:[k,h',w',C]
        q_k = CrossAttention(q=ldm_q, k=f_k, v=f_k);# q_k:[N,C]
        q_k = MLP(q_k);# q_k:[N,C]
        q_list.append(q_k)
    q_list = stack(q_list);# q_list:[K-1,N,C]
    return q_list
def decoder(z, first_f):
    # The decoder consists of a set of decoder layers, composed of Causal3DCNN, CrossAttention and up_scale layers.
    rec_video = repeat(first_f, K, dim=0);# [K, h', w', C]
    for _ in decoder_layers:
        # process and upsample features using Causal3DCNN and up_scale.
        rec_video = Causal3DCNN(rec_video)
        for k in range(1, K):
            rec_video[k] = CrossAttention(q=rec_video[k], k,v=z[k-1])
        rec_video = up_scale(rec_video)
    return rec_video;# rec_video:[T, h, w, C]
# Main Function
def dynamics_enhanced_latent_dynamics_model(video):
    # video:video sequences
    # extract change information from video frames
    z, first_f = encoder(video)
    z = FSQ(z);# quantize z using FSQ
    # dLDM training stage
    if is_train:
        # obtain reconstructed video frames.
        rec_video = decoder(z.detach(), first_f)
        loss = MSE(rec_video, video)
        denoising_loss = VDM(video, z, rec_video.detach())
        loss = rec_loss + denoising_loss
        return loss
    return z
```

Regarding task types (Fig. A.4), paper folding accounts for 55.3% (5.2 hours) of the total duration due to its longer execution time, while block building comprises 1.8 hours.

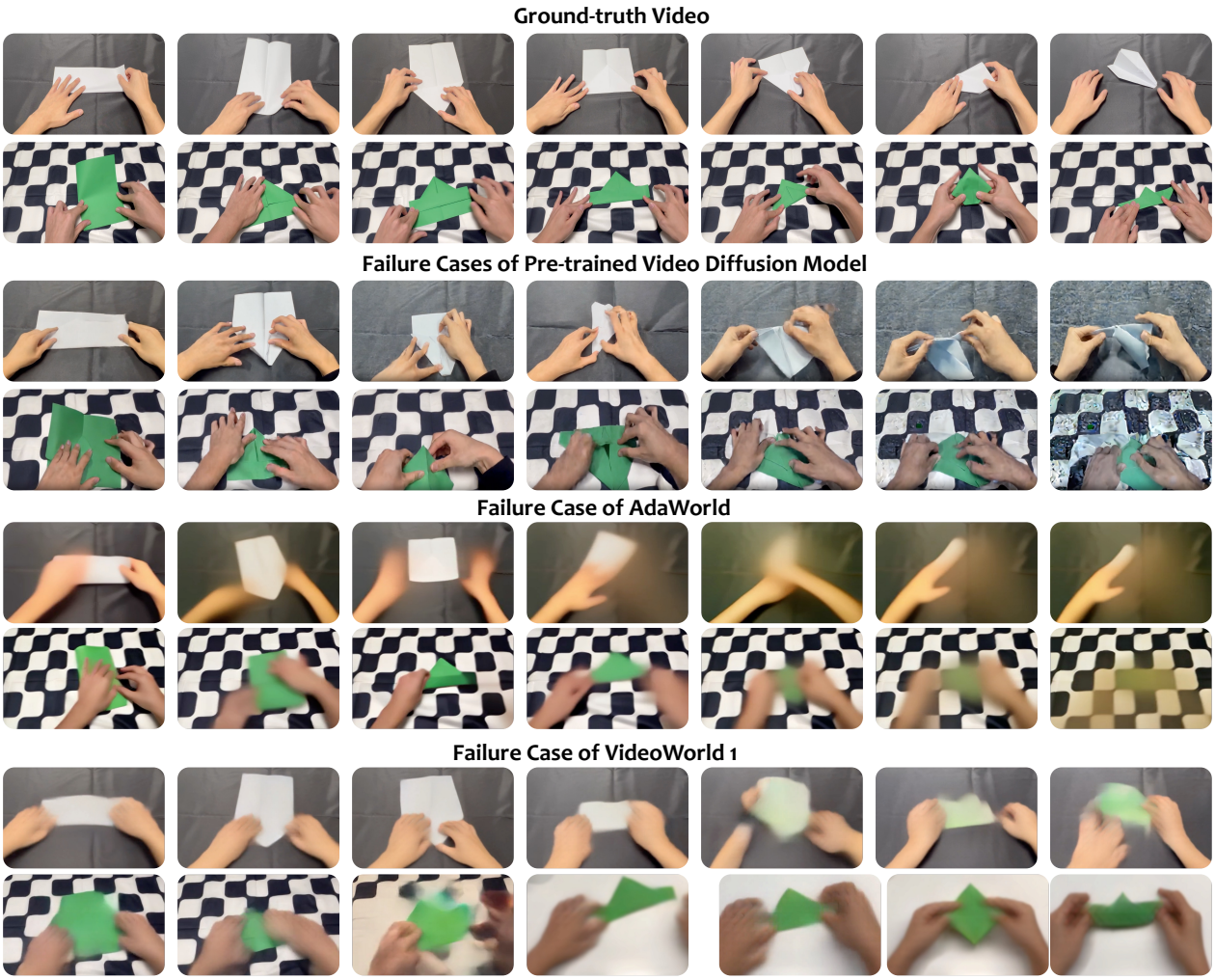

Figure B.7. **Visualization of failure cases** in other baselines on the Video-CraftBench. Best viewed in color.

Task classifier of Video-CraftBench. We detail the training of the classifier used to evaluate key step completion in Sec. 4.2. Its primary objective is to determine whether the model has reached key steps by analyzing paper shapes and block arrangements. Appearance distractions, such as texture, color, and the background drift shown in Fig. B.7 (last row), are disregarded. These visual quality is assessed separately via LPIPS and SSIM. Therefore, we initialize the classifier with DINOv2-Base (86M parameters), chosen for its exceptional geometric awareness and ability to extract fine-grained features.

To train this classifier, we construct a dataset of ∼25k labeled frames. First, we extract ∼15k frames of both key and non-key steps from the training and testing environments. To further enhance diversity and mitigate bias, we generate task execution videos using all models involved in the evaluation, manually verify successful trajectories, and extract an additional ∼10k frames. We then attach a classification head to the DINOv2 backbone and fine-tune the entire model. The classifier achieves a test accuracy of 96.1%, ensuring reliable judgment for subsequent evaluations.

D. Visualizations

Failure cases of other methods. Fig. B.7 illustrates failure cases for both large-scale pre-trained video generation models and latent action models on Video-CraftBench. As discussed in Sec. 5.3, although pre-trained video generation models may initially follow the correct action direction, their lack of disentanglement between action dynamics and appearance leads to rapid quality degradation and error accumulation over time. As a result, these models often stall after only a few steps or exhibit severe visual distortions.

For latent action models, we visualize AdaWorld in Fig. B.7. While AdaWorld preserves the overall scene appearance in unseen environments, its performance degrades substantially over long sequences, with clear declines in both visual quality and its ability to reach the correct key

Figure D.8. **More visualization of VideoWorld 2 on Video-CraftBench.**

steps. VideoWorld performs better than AdaWorld in reaching these key steps, but it still suffers from appearance drift that results in severe distortions in later frames.

More visualizations of VideoWorld 2. Fig. D.8 presents additional visualizations of VideoWorld 2 on Video-CraftBench. We also include more video demonstrations in the supplementary material. Thanks to the dLDM, VideoWorld 2 generates accurate long-horizon execution sequences in novel environments, producing visually coherent, high-quality videos. Note that the VDM processes only 93 frames at a time, while full task sequences can span thousands of frames. Long videos are therefore generated autoregressively by extending each segment from the final frame of the previous one. Because the VDM's inherent reconstruction noise accumulates over time, visual artifacts such as lighting, texture, or color shifts may gradually appear. Nevertheless, the key steps within the generated sequences remain accurate.