

# Ego-InBetween: Generating Object State Transitions in Ego-Centric Videos

## Supplementary Material

### 1. Overview

In this supplementary material, we first provide additional preliminaries of EgoIn, with a detailed explanation of the notation  $z_t$  used in Fig. 2 and Eq. 1 of the main paper. We further extend EgoIn to the Wan2.1-FLF2V-14B to validate its effectiveness on a Diffusion Transformer (DiT) architecture, referred to as EgoIn-DiT, and introduce the corresponding architectural preliminaries. Then, we present extensive additional experiments to further verify the effectiveness of EgoIn from multiple perspectives, **including the performance of EgoIn-DiT, additional ablation studies, generalization capability analyses, computational cost evaluations, generative controllability, and the integration of TransitionVLM with a commercial video generation model (Kling)**. We also provide more details on data curation, evaluation metrics, and visualizations showing the effect of the Object-aware Auxiliary Supervision. The detailed contents of this supplementary material are organized as follows.

- A - Preliminaries for video diffusion models with UNet and DiT architectures.
- B - Qualitative and quantitative comparison results of EgoIn-DiT against the original Wan2.1-FLF2V-14B.
- C - Additional ablation studies.
- D - Generalization capability analyses, including cross-benchmark evaluations, and inference using visual anchors extracted from YouTube videos to validate generalization to real-world visual inputs.
- E - Computational cost evaluations.
- F - Evaluation with six additional metrics.
- G - Demonstrations of EgoIn’s controllability across three challenging state transition scenarios.
- H - Evaluation of the proposed TransitionVLM integrated with a commercial video generation model (Kling).
- I - Additional details of the data curation process.
- J - Additional details of the evaluation metrics used in the main paper.
- K - Visualization of multi-frame object localization during state transitions at inference.
- L - Discussion of limitations and future research directions.
- M - Video demonstrations are provided in the file named “supp video demo”.

### 2. PRELIMINARY: Video Diffusion Models

Diffusion-based generative models have recently become prominent in video synthesis, defining a forward stochastic process that perturbs clean samples  $x_0 \sim p_{\text{data}}(x)$  into progressively noisier representations  $x_t$ , eventually yielding  $x_T \sim \mathcal{N}(0, I)$ . This forward process  $q(x_t|x_0, t)$  comprises  $T$  discrete timesteps, each adding incremental noise to the input. The reverse or denoising process  $p_\theta(x_{t-1}|x_t, t)$  employs a neural network  $\epsilon_\theta(x_t, t)$  trained to predict the injected noise using:

$$\min_{\theta} \mathbb{E}_{t, x \sim p_{\text{data}}, \epsilon \sim \mathcal{N}(0, I)} \|\epsilon - \epsilon_\theta(x_t, t)\|_2^2. \quad (1)$$

We build our method upon a video Latent Diffusion Model (LDM) [10], DynamiCrafter [32]. Given an input video  $x \in \mathbb{R}^{N \times 3 \times H \times W}$ , each frame is encoded into a latent representation  $z = E(x)$ . The diffusion and denoising processes are performed in the latent space via  $z_t = p(z_0, t)$  and  $z_{t-1} = p_\theta(z_t, c, t)$ , where  $c$  denotes conditioning. Following [32], the conditions include the text description  $c_T$  and the first and last frames  $c_I^1$  and  $c_I^N$ . The objective becomes:

$$\min_{\theta} \mathbb{E}_{E(x), t, \epsilon \sim \mathcal{N}(0, I)} \|\epsilon - \epsilon_\theta(z_t, t, \text{fr}, c_I^1, c_I^N, c_T)\|_2^2. \quad (2)$$

where fr denotes the frame-rate control. In our formulation, EgoIn introduces frame-wise transition conditions  $\{\tilde{F}_i^*\}_{i=1}^N$  to guide the generative trajectory. This yields the modified training objective:

$$\min_{\theta} \mathbb{E}_{E(x), t, \epsilon \sim \mathcal{N}(0, I)} \left\| \epsilon - \epsilon_\theta(z_t, t, \text{fr}, \{\tilde{F}_i^*\}_{i=1}^N) \right\|_2^2. \quad (3)$$

Beyond UNet-based latent diffusion models, Diffusion Transformer (DiT) architectures adopt a flow-matching formulation followed by rectified flow (RF). Given a clean latent  $\mathbf{x}_0$  and Gaussian noise  $\epsilon \sim \mathcal{N}(0, I)$ , the noisy latent at continuous timestep  $t \in [0, 1]$  is obtained by linear interpolation:

$$\mathbf{x}_t = (1 - t)\mathbf{x}_0 + t\epsilon. \quad (4)$$

The model learns a time-dependent vector field  $v_\theta(t, \mathbf{x}_t)$  via the Conditional Flow Matching objective:

$$\mathcal{L}_{\text{flow-matching}} = \mathbb{E}_{t, \epsilon, \mathbf{x}_0} \|v_\theta(t, \mathbf{x}_t) - (\epsilon - \mathbf{x}_0)\|_2^2. \quad (5)$$

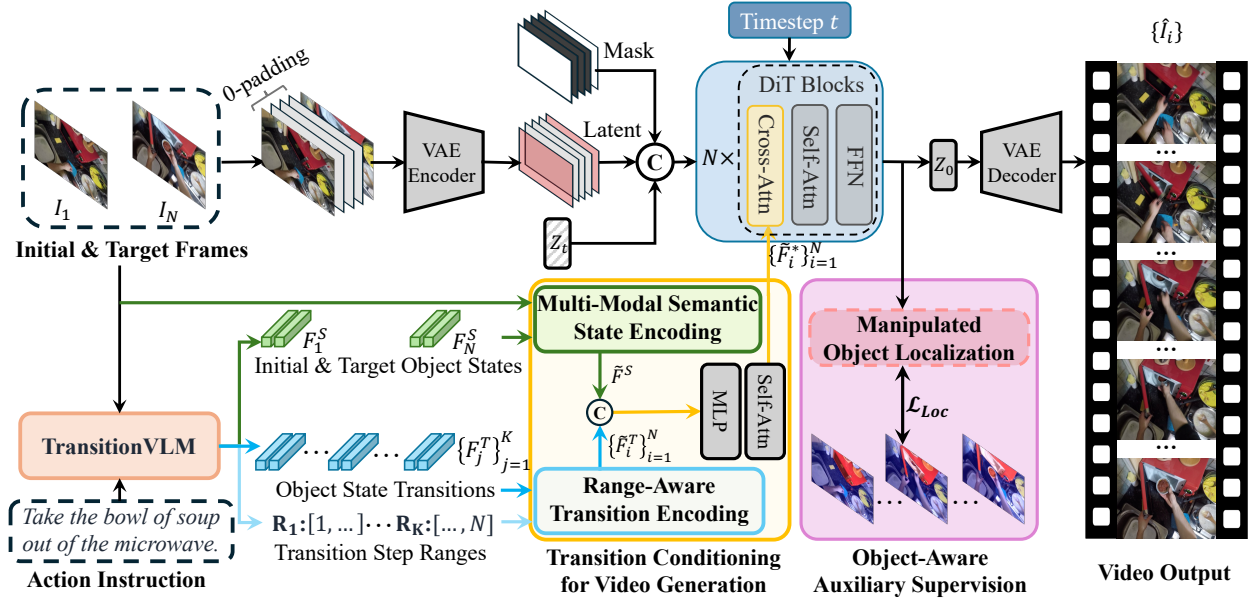


Figure 1. Illustration of extending the proposed EgoIn to a DiT-based diffusion model. Unlike UNet architectures that inject conditioning through multi-scale spatial features, the DiT design modulates latent tokens within transformer blocks, enabling globally contextualized and semantically coherent state transitions.

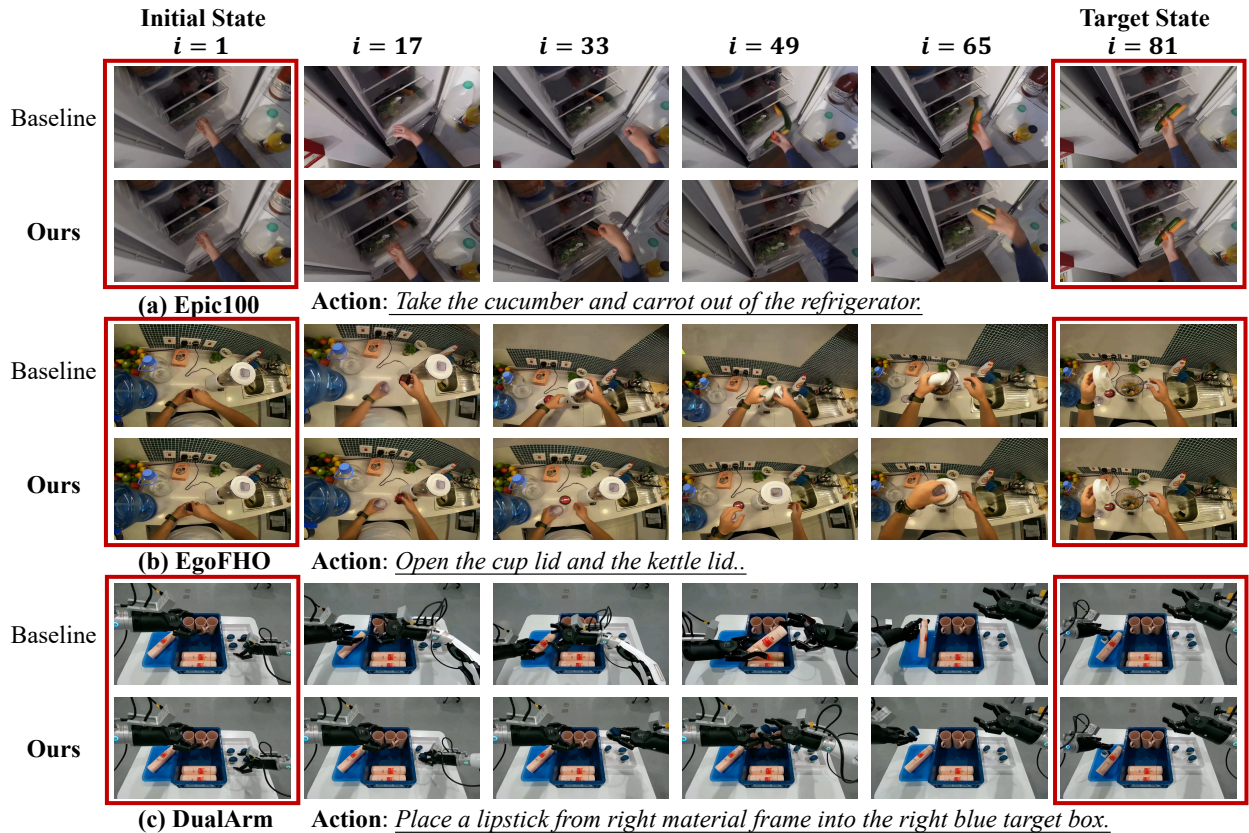


Figure 2. Qualitative comparison between the baseline model (Wan2.1-FLV2V) and our EgoIn-DiT on the Epic100, EgoFHO, and DualArm datasets. Intermediate frames ( $i = 17, 33, 49, 65$ ) from the generated sequences are shown.

Method	Epic100				EgoFHO				DualArm			
	FVD↓	VTQ↑	VTC↑	VIC↑	FVD↓	VTQ↑	VTC↑	VIC↑	FVD↓	VTQ↑	VTC↑	VIC↑
Wan2.1-FLF2V	226.91	0.9053	0.2284	0.9195	234.72	0.9004	0.2191	0.9248	223.84	0.9118	0.2272	0.9373
<b>EgoIn-DiT (Ours)</b>	<b>186.75</b>	<b>0.9177</b>	<b>0.2422</b>	<b>0.9347</b>	<b>193.58</b>	<b>0.9149</b>	<b>0.2383</b>	<b>0.9394</b>	<b>188.30</b>	<b>0.9217</b>	<b>0.2435</b>	<b>0.9450</b>

Table 1. Quantitative comparison between Wan2.1-FLF2V and EgoIn-DiT on the Epic100, EgoFHO, and DualArm datasets, reporting FVD, VTQ, VTC, and VIC scores.

This continuous-time formulation allows DiT models to capture long-range spatio-temporal dependencies while benefiting from the stability and efficient training dynamics of rectified-flow parameterizations.

### 3. More Experimental Results.

#### 3.1. Evaluation of EgoIn-DiT

We select Wan2.1-FLF2V-14B [29] as the foundation model for EgoIn-DiT because it provides strong conditioning capabilities on the first and last frames and can generate videos with realistic physical dynamics and large motion. However, its ability to understand and reason about the fine-grained actions and corresponding object state changes between the two given states is limited. This often leads to motion disruptions, abrupt state inconsistencies, or objects unexpectedly disappearing or appearing mid-sequence, making it unsuitable for directly modeling state transitions between the initial and target frames. As shown in Fig. 2, our EgoIn-DiT effectively equips the model with the ability to reason and generate coherent transformations between the start and target states. The predicted transitions are semantically meaningful, and the generated motion exhibits temporal consistency and smoothness. We further conduct quantitative evaluations using the same metrics as in the main paper, including FVD, VTQ, VTC, and VIC. As reported in Tab. 1, EgoIn-DiT consistently outperforms the baseline, which is obtained by fine-tuning the Wan2.1-FLF2V-14B model on the corresponding dataset, across all evaluation metrics.

**Training details.** We train EgoIn-DiT using 81-frame videos at a resolution of  $480 \times 832$  in two stages. In the first stage, we optimize the TC module for 10k steps with a learning rate of  $1 \times 10^{-4}$  while keeping all other parameters frozen, allowing the TC module to align with the conditional space of the DiT. In the second stage, we introduce LoRA modules with `lora_rank=128` and `lora_alpha=64` into the DiT blocks and jointly fine-tune them with the TC module for another 10k steps using a learning rate of  $5 \times 10^{-5}$ .

#### 3.2. Additional Ablation Study

**Ablation on VLMs for state transition modeling.** We compare different VLMs for modeling the transition process, including the original Qwen2.5-VL [2], GPT-4o [1], and the proposed TransitionVLM. For Qwen2.5-VL, we

VLM	Epic100			
	FVD↓	VTQ↑	VTC↑	VIC↑
Qwen2.5-VL	269.32	0.8871	0.2141	0.9160
GPT-4o	232.84	0.9017	0.2308	0.9255
<b>TransitionVLM (Ours)</b>	<b>215.27</b>	<b>0.9081</b>	<b>0.2373</b>	<b>0.9313</b>

Table 2. Quantitative comparison of state transition video generation using different VLMs.

extract text features before the prediction heads. For the GPT-4o, we use the text encoder of DynamiCrafter to convert their textual outputs into features. All VLM-derived transition features are fed into the Transition Conditioning (TC) module in the same manner as TransitionVLM. EgoIn is then finetuned individually with each type of input. The results are presented in Tab. 2. Although the GPT-4o demonstrate strong contextual reasoning and structural interpretation capabilities, it tend to produce plausible but incorrect descriptions in specialized domains. As discussed in the main paper, prompting GPT-4o with only the initial and target frames frequently leads to redundant, vague, or task-irrelevant descriptions, which can be mitigated by providing additional contextual information from the original video. TransitionVLM addresses this limitation by being trained on GPT-4o outputs generated with richer contextual cues from the original videos, enabling it to produce more reliable transition descriptions even when only two visual anchors are provided at inference. This, in turn, enhances the video diffusion model’s ability to understand and reason about object state transitions, leading to consistent improvements across all evaluation metrics. **Our data curation pipeline can be easily adapted to different GPT models, and we believe that more advanced models will further improve EgoIn.**

**Ablation on the transition conditions.** We conduct this experiment using four types of transition conditions for video generation: (1) textual-level instruction of the input simple action  $T_A$ ; (2) textual-level state and transition instructions  $T^S, T^T$  generated by the original VLM [2]; (3) textual-level state and transition instructions  $T^S, T^T$  generated by TransitionVLM; and (4) feature-level state and transition representations  $F^S, F^T$  extracted from the linear head of TransitionVLM. These correspond to rows 1–4 in Tab. 3. Incorporating additional textual instructions that describe the state and transition steps improves the model’s performance (row 2 vs. row 1). Using the fine-tuned Tran-



VLM Tuning	Inputs	Epic100			
		FVD↓	VTQ↑	VTC↑	VIC↑
×	$T_A$	283.47	0.8837	0.2078	0.9118
×	$T^S, T^T$	269.32	0.8871	0.2141	0.9160
✓	$T^S, T^T$	232.79	0.9004	0.2282	0.9252
✓	$F^S, F^T$	<b>215.27</b>	<b>0.9081</b>	<b>0.2373</b>	<b>0.9313</b>

Table 3. Quantitative comparisons of ablation variants using different textual conditions.

Method	Epic100		
	Reason. ↑	Align. ↑	Motion. ↑
w/o Range	10.15%	15.18%	11.69%
w/ Uniformly Distributed Range	31.89%	40.03%	34.77%
w/ Predicted Range (ours)	57.96%	44.79%	53.54%

Table 4. User preference rates to evaluate the necessity of transition temporal range in the reasonability of transition steps (Reason.), their alignment with the instruction (Align.), and motion quality (Motion.).

sitionVLM leads to more accurate visual reasoning, and its generated instructions further enhance transition quality (row 3 vs. row 2). Moreover, using feature-level representations provides greater robustness and preserves richer semantic information than directly using raw textual outputs from the VLM, resulting in further improvement for video diffusion model (row 4 vs. row 3).

**Ablation on the temporal range of transition.** We conduct a user study to evaluate the results generated with different temporal ranges. This is because the temporal range of transitions is subjective and highly sensitive to human perception, making it difficult to assess using existing metrics. The results are presented in Tab. 4. The baseline, "w/o Range," removes the Range-Aware Transition Encoding from EgoIn, meaning the transition conditions are no longer frame-wise. Our method received greater user preference, with over 50% of the votes, outperforming the results generated using other temporal ranges.

### 3.3. Generalization Capability of the EgoIn.

**Cross-dataset evaluation.** We include cross-dataset generalization results for EgoFHO  $\rightarrow$  Epic100 and Epic100  $\rightarrow$  EgoFHO in Tab. 5. Our model demonstrates strong generalization ability and outperforms previous methods that are trained and evaluated on the same dataset (Tab. 1, Main Paper).

**Real-world visual anchors.** We further evaluate the generalization capability of the EgoIn using visual anchors extracted from YouTube videos. Specifically, we collect several YouTube videos which include unseen initial and target states. From each video, we extract two frames with a long temporal span as the initial and target states, and then provide a simple action instruction based on these anchors. As shown in Fig. 3, EgoIn generates coherent and reason-

able video results. For example, in Fig. 3(a), the initial state shows a red piece being held, with various components such as rods and gears arranged on the table, while the target state shows a yellow rod secured in place, completing an assembly step. EgoIn produces a smooth transition in which the model retrieves the yellow rod from the table, brings it toward the red piece, and inserts it to complete the assembly. These results demonstrate the strong generalization capability of our method and highlight its potential for applications in areas such as robotic manipulation, computer-aided design, and embodied AI.

Training	Epic100				EgoFHO			
	FVD↓	VTQ↑	VTC↑	VIC↑	FVD↓	VTQ↑	VTC↑	VIC↑
<b>Epic100</b>	215.27	0.9081	0.2373	0.9313	239.49	0.8904	0.2266	0.9329
<b>EgoFHO</b>	244.10	0.8963	0.2264	0.9224	203.85	0.8987	0.2340	0.9396

Table 5. Generalization results across the Epic100 and EgoFHO datasets using models trained on the other dataset.

### 3.4. Computational Cost Analysis.

We evaluate the computational cost of EgoIn and the baseline model DC-Interp [32] on a single MI250 GPU without acceleration libraries such as vLLM [13]. All models are configured with 50 inference sampling steps to generate a 16-frame transition video at a resolution of 320×512. For inference latency, we measure the average runtime over 50 runs, as reported in Tab. 6. Compared to DC-Interp, EgoIn introduces only a small latency increase of 4.4%, which is primarily attributed to the additional TransitionVLM inference. Regarding GPU memory usage, the incorporation of a 7B VLM leads to an additional consumption of 16.7 GB compared to the baseline.

### 3.5. Comparison with Six Additional Metrics

To comprehensively assess the effectiveness of the proposed EgoIn, we provide six additional evaluation metrics based on EvalCrafter [19]:

**LPIPS** To assess the perceptual similarity between generated transition video and ground-truth video, we evaluate the distance between image patches. The lower score indicates greater similarity to the ground truth.

**Inception Score (IS).** To assess the video quality, we calculate the inception score [23], using a pre-trained Inception Network [26]. A higher inception score indicates more diverse generated content.

**Warping Error (WE).** We adopt the warping error, a metric widely used in previous blind temporal consistency methods [14, 15], to measure temporal consistency. In detail, we use a pre-trained optical flow estimation network [27] to compute the optical flow between two frames. The pixel-wise warp differences are then calculated and averaged across all frame pairs of the generated frames.

Method	Time	Memory
DC-Interp	45s	18.5GB
EgoIn(Ours)	47s	35.2GB

Table 6. Comparison of the proposed EgoIn model and DC-Interp [32] in computational efficiency and memory cost.

**BLIP-BLEU.** To evaluate the alignment between input prompt and generated frames, we first adopt BLIP2 [16] for generating a caption of the generated video and then use BLEU [21] to calculate the text alignment score.

**Perceptual Input Conformity (PIC).** Inspired by [32], we compute the perceptual distance between the input frames (initial and target frames) and the generated frames using the diffusion-based metric DreamSim [7]. The calculated scores are then averaged to obtain the PIC.

**Perceptual Video Conformity (PVC).** We also use a variant of PIC to calculate the mid-level similarities between the generated frames and the ground truth at each time step, considering various evaluation aspects such as image layout, object poses, and semantic content. The scores for each frame are then averaged to compute the PVC score, which provides a better alignment with human perception.

The quantitative comparison results between the proposed EgoIn and three SOTA methods [4, 32, 33] are shown in Tab. 7. Our method outperforms the other SOTAs on both the Epic100 and EgoFHO benchmarks, with the only exception being the WE metric, where FILM achieves a higher score. FILM tends to generate linearly smooth transitions, whereas EgoIn is able to model larger and more dynamic state changes.

### 3.6. Generative Controllability of the EgoIn.

We evaluate the controllability of the proposed EgoIn across three challenging state transition scenarios: (1) Transition processes corresponding to different action instructions and the same visual anchors; (2) Transition processes corresponding to the same action instructions and the initial frame but different target state; (3) Transition processes corresponding to the reversed action instruction and visual anchors.

**Different action instructions, same visual anchors.** The results are illustrated in Fig. 4(a). EgoIn reaches the same target state from the same initial state by following different transition steps based on two different action instructions. This demonstrates that the proposed EgoIn is equipped with diverse generation capabilities.

**Same action instructions, same initial frame, different target states.** The results are presented in Fig. 4(b). EgoIn generates distinct transition steps corresponding to different target states. This demonstrates that the EgoIn does not over-rely on textual instructions but also effectively incorporates visual information during video generation.

**Reversed action instruction and visual anchors.** The results are provided in Fig. 4(c). In one case, the initial state depicts a closed microwave door, while the final state shows it open. EgoIn successfully generates the key intermediate steps: (1) The gripper secures the handle of the microwave door. (2) The door is slightly opened, and the gripper moves around to the other side. (3) The gripper pushes outward from the other side, fully opening the microwave oven door. For the reversed case, EgoIn generates a different yet reasonable transition sequence. This validates that our model possesses a “reverse thinking”-like capability.

### 3.7. Comparison with Single-Frame Conditioned Video Generative

To highlight the proposed EIVST task, we compare our method with single-frame-conditioned video generative models, specifically Seer [9] and DC-I2V [32], whose weights were tuned on our dataset for a fair comparison. Fig. 5 presents the qualitative comparison results. However, these methods face three challenges: (1) difficulty ensuring that the generated video reaches the desired target state, (2) generating incorrect object details when the given image lacks necessary information (e.g., placing food in the fridge, but its contents are unseen in the initial frame), and (3) limited actions throughout the transition sequence. Compared to that, the proposed EgoIn, with its strong visual reasoning capability, accurately generates state transitions from the initial state to the desired target state, offering enhanced controllability in video generation. Additional qualitative results of the proposed EgoIn are shown in Fig. 6.

### 3.8. Kling with TransitionVLM

In this experiment, we evaluate Kling by providing it with the same two visual anchors but different textual prompts. As a baseline, we use the action instruction directly as the textual prompt, as shown in Fig. 7(a). Kling produces visually pleasing videos with high quality and smooth motion. However, its transition process omits key actions such as “flip”. We further enhance the textual prompt by incorporating additional semantic information. Specifically, we concatenate the action instruction with transition steps generated by Qwen2.5VL-7B[2]. As shown in Fig. 7(b), this improved textual prompt enables Kling to generate additional transition steps. However, directly using Qwen2.5VL-7B for our task is not ideal, as it tends to generate plausible-sounding actions that lack factual accuracy, e.g., incorrect action “adjust its position”. This causes inconsistencies in the bowl’s appearance across frames. In contrast, the proposed TransitionVLM generates reasonable transition steps, including the “flip the bowl” action, leading to smoother and more coherent transitions. As shown in Fig. 7(c), this improvement allows Kling to produce transitions that more accurately reflect real-world object interactions. These re-

Method	Epic100						EgoFHO					
	LPIPS ↓	IS ↑	WE ↓	BLIP-BLEU ↑	PIC ↑	PVC ↑	LPIPS ↓	IS ↑	WE ↓	BLIP-BLEU ↑	PIC ↑	PVC ↑
FILM[22]	0.3281	4.1491	<b>0.0069</b>	0.0255	0.8076	0.8361	0.3271	5.6059	<b>0.0056</b>	0.1087	0.8461	0.8458
SEINE[4]	0.3701	5.1443	0.0225	0.0265	0.8198	0.8445	0.3481	6.7173	0.0200	0.1097	0.8569	0.8667
DC-Interp[32]	0.3643	4.9903	0.0171	0.0258	0.8237	0.8524	0.3640	6.5538	0.0184	0.1099	0.8478	0.8576
<b>EgoIn(Ours)</b>	<b>0.3136</b>	<b>5.2058</b>	0.0110	<b>0.0297</b>	<b>0.8305</b>	<b>0.8733</b>	<b>0.3223</b>	<b>6.8099</b>	0.0159	<b>0.1165</b>	<b>0.8580</b>	<b>0.8657</b>

Table 7. Quantitative comparison with text-guided state-of-the-art video completion methods on Epic100 [5] and EgoFHO [8]. The scores of the LPIPS, IS, WE, BLIP-BLEU, PIC and PVC are reported.

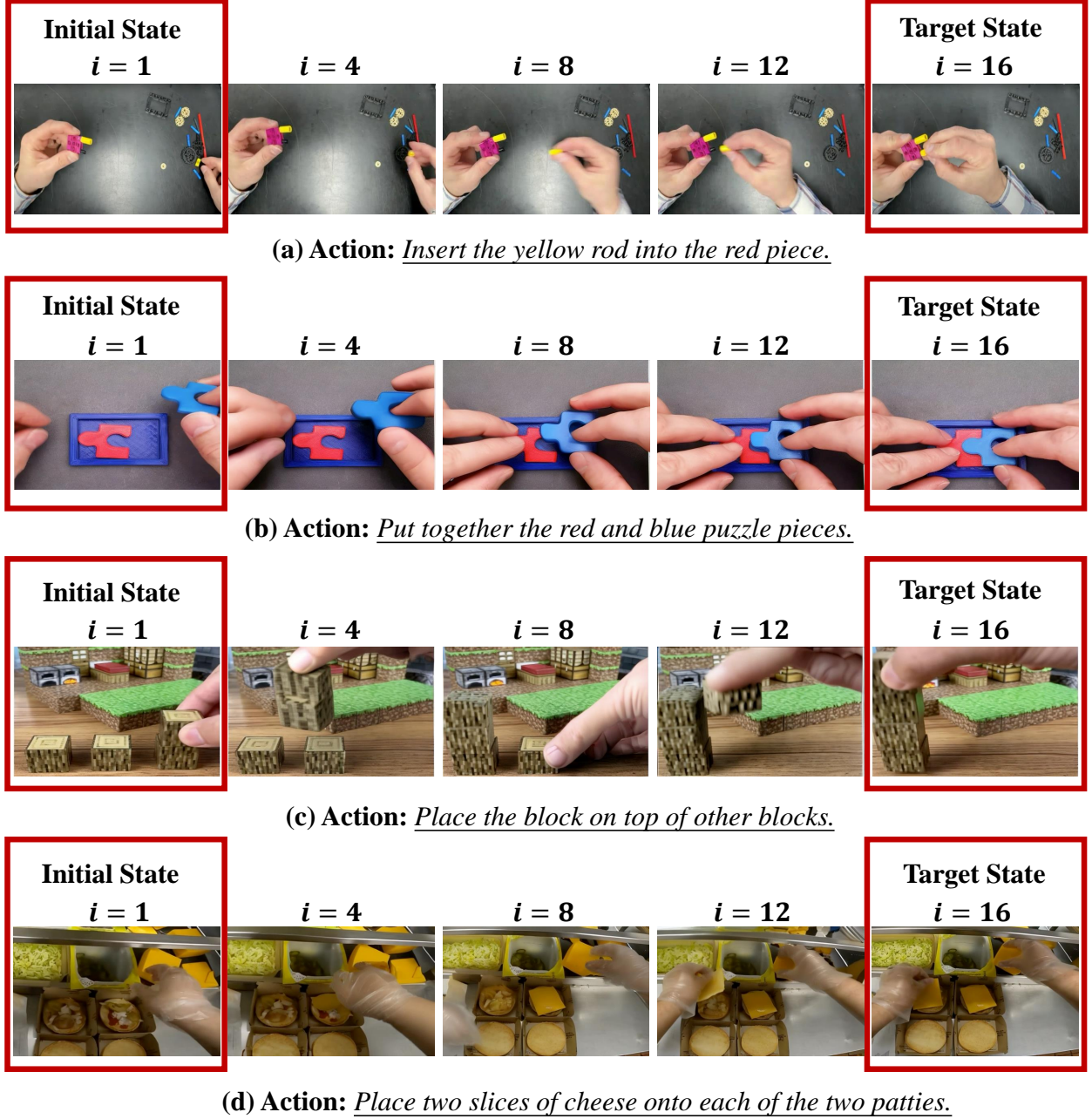


Figure 3. Transition generation for unseen objects, states, and scenarios.



sults demonstrate that TransitionVLM enhances video generation models by enabling them to handle complex transitions where visual reasoning is essential.



Figure 4. Generated state transitions in three challenging scenarios.

## 4. More Details

### 4.1. Data Curation for Transition Video Datasets.

Most existing egocentric video datasets [12, 17, 18, 25] are primarily designed for action or activity understanding, and thus overlook their potential for generation-oriented

tasks such as visual state transition modeling, which requires minimal scene and viewpoint changes, as object state changes appear differently under varying viewpoints. To the best of our knowledge, there is no available large-scale and transition-aware dataset for video generation in ego-centric views. Such a dataset is crucial for bridging the gap between human cognition and machine understanding of object state transformations. To construct it, we leverage existing hand-object and robot-object interaction datasets [5, 6, 8, 20, 24, 31] and automatically segment the original video clips into finer-grained sequences based on annotation information, with each sequence corresponding to an object state transition triggered by an instructed action. To remove clips containing static scenes or excessive camera motion, we compute the frame similarity between the first and last frames and the average optical flow for each segmented clip, discarding those without clear object state changes or those affected by large viewpoint shifts, so that the model can better focus on object-centric transitions. We further assess the visual quality of each clip by evaluating brightness, blurriness, and noise levels, and filter out samples with low-quality visual input. The resulting curated dataset contains diverse state transitions across indoor and outdoor environments. For each fine-grained clip, we generate a concise action instruction using QwenVL-2.5 [2], replacing the coarse activity annotation provided for the original long video.

**Text prompt used for data curation in GPT-4o.** Fig. 8 illustrates the detailed prompts used for GPT-4o during data curation. Additional guidance information (box and full frames) cannot be used during inference.

### 4.2. Explanation of The Evaluation Metrics

In the main paper, we assess the generated object state transition from two aspects: video quality and video-condition consistency.

**Video Quality.** We use the Fréchet Video Distance (FVD) [28] and the temporal quality score in VBench[11], denoted as “TVQ”, which is calculated as a weighted average of 5 dimensions, including dynamic degree, motion smoothness, temporal flickering, subject consistency and background consistency.

**Video-condition Consistency.** We assess video-condition consistency from two perspectives. The first is the video-text consistency. Specifically, we calculate cosine similarity between the ViCLIP[30] features of the generated video and action instruction and denote it as “VTC”, following the approach for overall consistency in VBench[11]. The second is the consistency of the generated video and given initial and target state frames. We extract features using DINOv1[3] and DreamSim[7], and calculate similarity as subject and background consistency, respectively. Then the overall video-image consistency is the weighted aver-

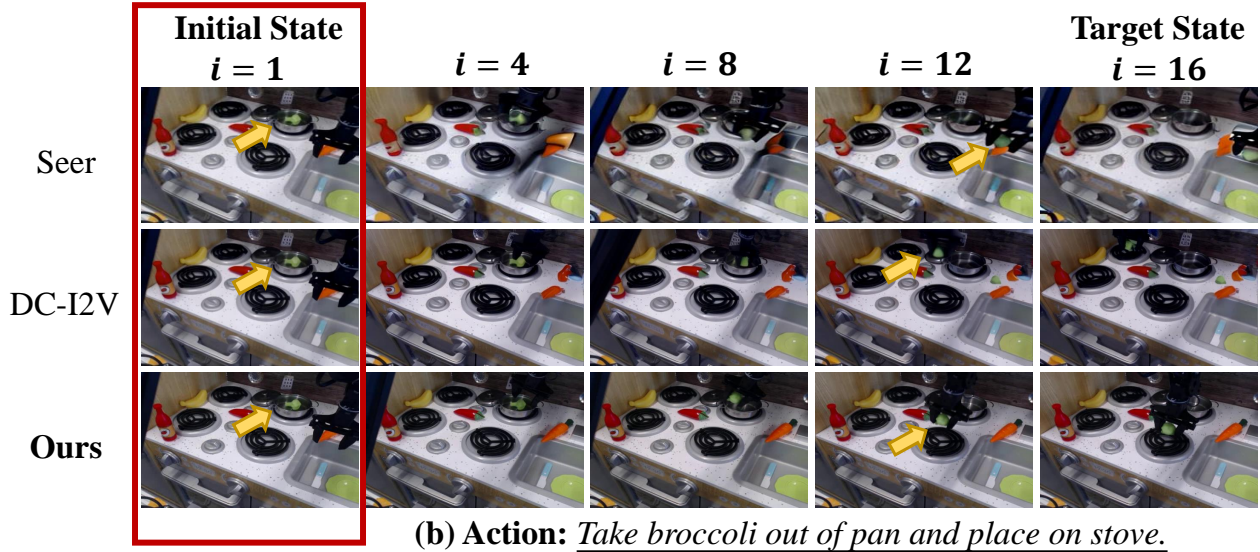
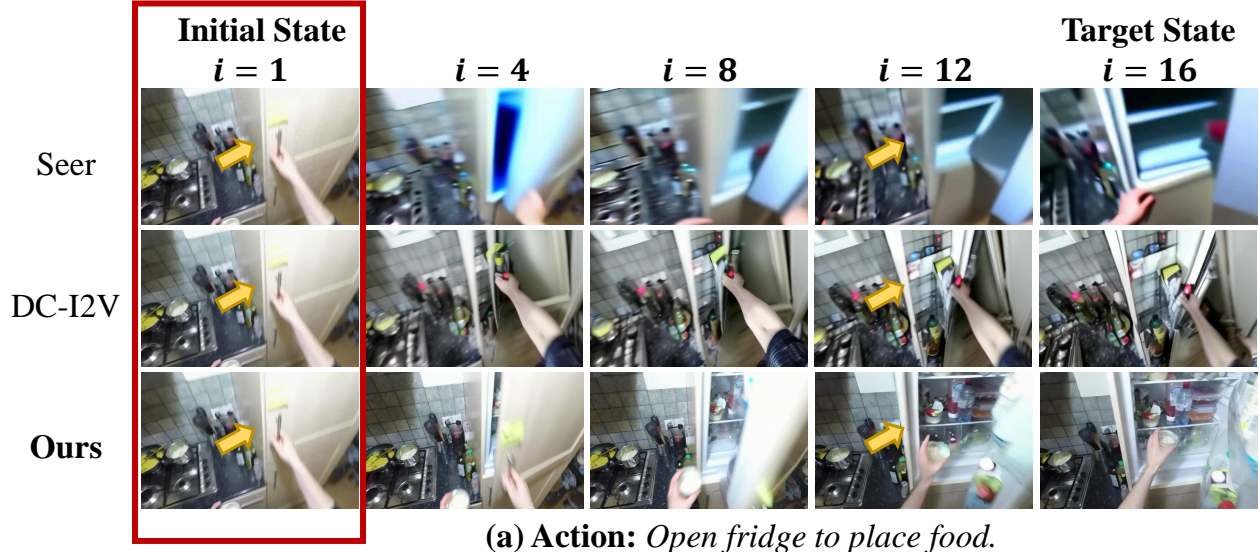


Figure 5. Qualitative comparison of the proposed EIVST with two single-frame-based video generation methods that use the initial frame and action instruction as condition.

age of them, and denoted as “VIC”, following the setting of VBench[11].

#### 4.3. Implementation Details of Object-aware Auxiliary Supervision.

Our auxiliary loss is computed using the **clean video latent**, estimated from the noisy input  $Z_t$  via the UNet-predicted noise  $\epsilon_t$ . This latent is then passed to a localization head consisting of two stacked 3D convolutional layers with kernel size  $3 \times 3 \times 3$ . State transitions induce spatiotemporal variations that are effectively captured by 3D convolutions, which also enable the localization of different key

objects involved in the transition process. The resulting feature maps are supervised using a downsampled ground-truth mask, where pixels within the key object region are labeled as 1 and all others as 0. A mean pixel-wise cross-entropy loss is applied to encourage the localization head to assign high probabilities to key object regions.

We visualize the predicted masks for intermediate frames using the localization head described in Section 3.4 of the main paper. As illustrated in Fig. 9, the predicted masks consistently focus on the manipulated objects across multiple frames (highlighted by the **green arrow**) and closely align with the actual object shapes, even in cases



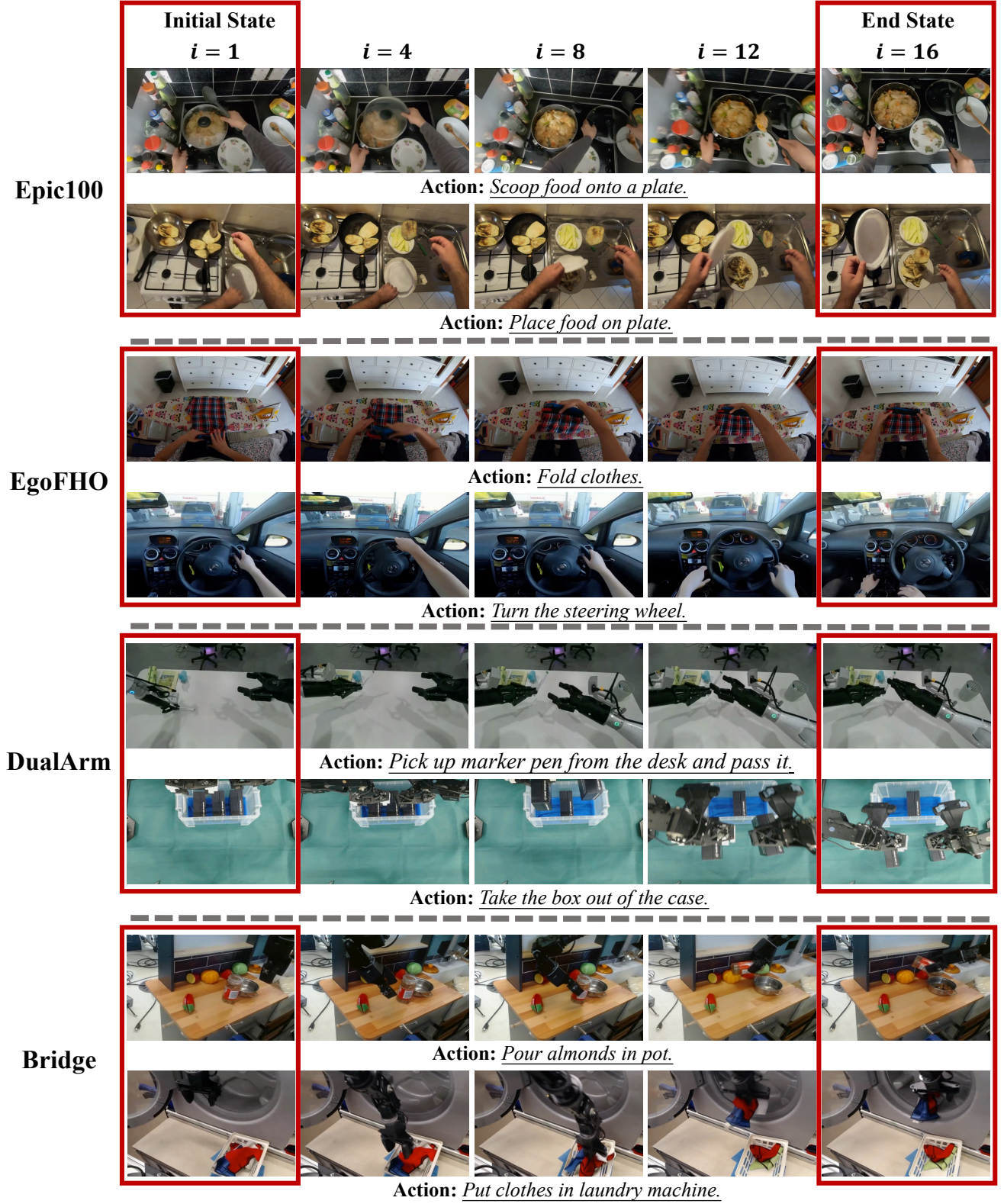


Figure 6. Additional qualitative results of the proposed EgoIn on the Epic100 [5], EgoFHO [8], DualArm [20, 24, 31], and Bridge [6] datasets.

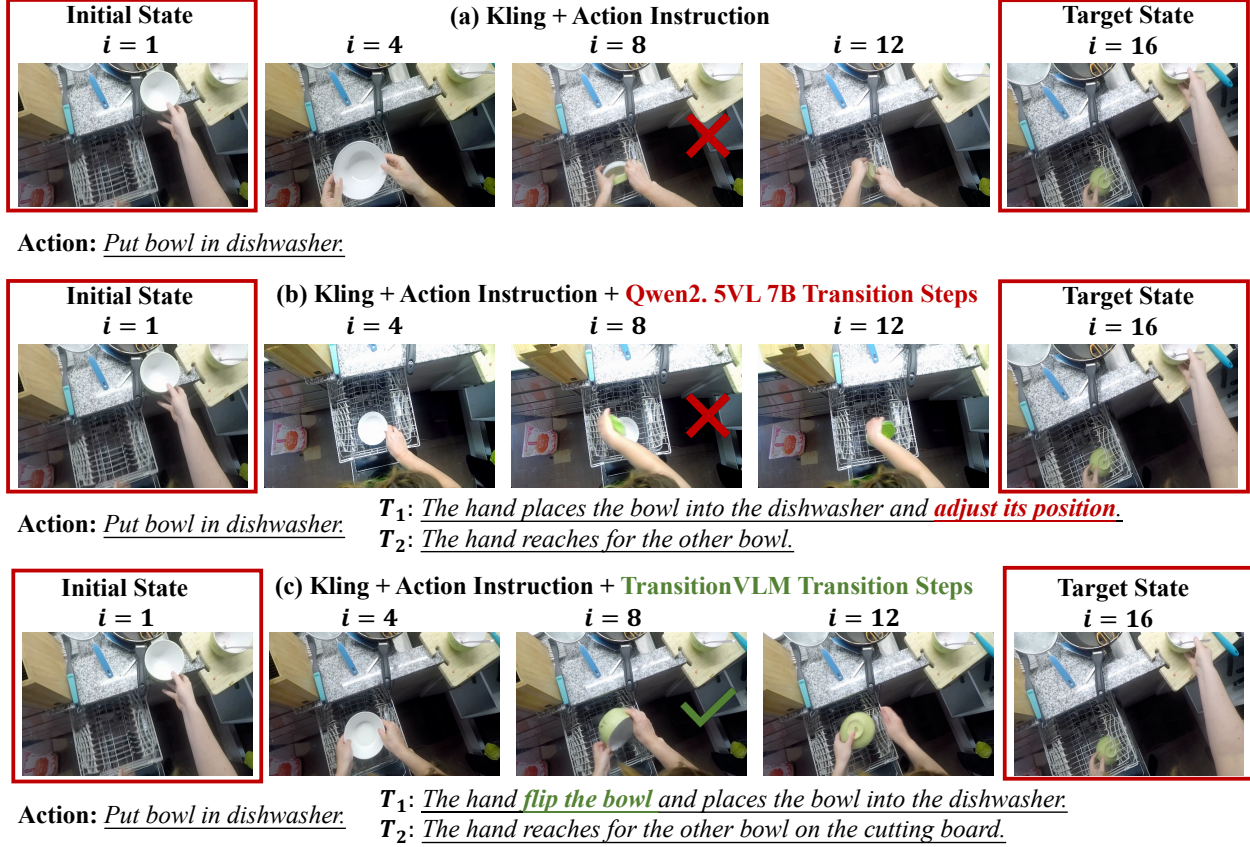


Figure 7. Transition videos generated using Kling with different textual prompts.

#### Prompt of GPT-4o for initial and target object states

**Input:** Given two images representing the initial and final frames of an action sequence, along with an overall process description  $T$ : "**{Action Instruction}**". For each image, additional spatial information is provided in the form of bounding boxes for objects that may interact with the hand. Each bounding box is represented by four numbers in the format:  $[x\text{-coord of top-left}, y\text{-coord of top-left}, x\text{-coord of bottom-right}, y\text{-coord of bottom-right}]$ . The coordinate origin is at the top-left corner of the image, with the x-axis running horizontally along the top and the y-axis running vertically along the left. All coordinates are normalized to the range  $[0, 1]$ .

**Task:** Compare the two images and the provided bounding boxes of potentially manipulated objects to identify regions where object state changes may have occurred. Then, referring to the semantics of the textual instruction describing the process, infer the category of the manipulated object. Based on this information, provide a description of the object's state in the initial frame (before the action occurs) and in the final frame (after the action).

**Output:** Describe the object state in each image. The descriptions should ignore irrelevant objects and focus only on the hand and the manipulated object.

#### Prompt of GPT-4o for object state transitions and transition step ranges

**Inputs:** You are given a 16 keyframes extracted in order from a video showing a manipulation process. Specifically, the first frame corresponds to the beginning of the process, the last frame corresponds to its completion, and the intermediate keyframes represent the steps in between. Additionally, you are provided with an overall process description  $T$ : "**{Action Instruction}**".

**Task:** Your task is to divide the 16 keyframes into 1 to 4 key steps, where each step corresponds to a logical subprocess, and they are arranged in order. Furthermore, the predicted key steps follow the constraints of the first and last frames, and do not output steps that occur before the first frame or after the last frame. The given overall process description should be referenced.

**Outputs:** For each key step, provide a text description of what is happening between hands and objects, and corresponding object state transition. Then based on the predicted key steps, split given 16 keyframes into sequential and non-overlapping frame ranges. Here are some examples to guide response: *Examples for learning: (1) Example-1 (2) Example-2 ... (M) Example-M*

Now, following the above format, predict split key steps with corresponding description and frame range according to given inputs.

Figure 8. The prompts of GPT-4o in data curation for VLM tuning.

involving large motion. Note that this is not used during inference; we present it here solely to validate its effectiveness.

#### 4.4. Limitation and Future Work.

Although EgoIn performs well across diverse egocentric transitions on both Unet-based and DiT-based architectures, generating long-horizon state transitions that involve sub-

stantial scene or viewpoint changes remains a challenging problem. Addressing such complex transitions requires more robust modeling of long-range dependencies and multi-view consistency. We view this as an important direction for future work and plan to further enhance our framework to better handle large viewpoint shifts and dynamic environments.

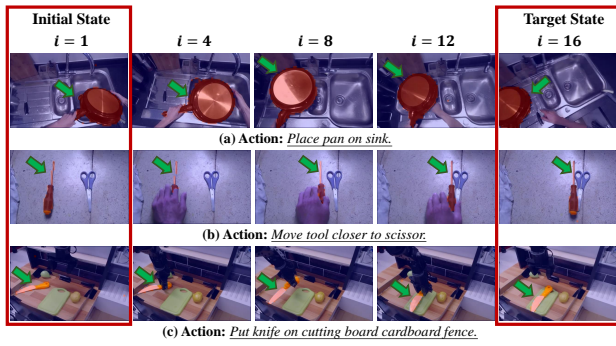


Figure 9. Visualization of predicted multi-frame masks of the manipulated objects and the generated videos. (**Zoom-in for best view**)



## References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *CoRR*, abs/2303.08774, 2023. 3
- [2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *CoRR*, abs/2502.13923, 2025. 3, 5, 7
- [3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 7
- [4] Xinyuan Chen, Yaohui Wang, Lingjun Zhang, Shaobin Zhuang, Xin Ma, Jiashuo Yu, Yali Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Seine: Short-to-long video diffusion model for generative transition and prediction. In *ICLR*, 2023. 5, 6
- [5] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018. 6, 7, 9
- [6] Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *CoRR*, abs/2109.13396, 2021. 7, 9
- [7] Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. *CoRR*, abs/2401.09985, 2023. 5, 7
- [8] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022. 6, 7, 9
- [9] Xianfan Gu, Chuan Wen, Weirui Ye, Jiaming Song, and Yang Gao. Seer: Language instructed video prediction with latent diffusion models. In *ICLR*, 2024. 5
- [10] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *CoRR*, abs/2210.02303, 2022. 1
- [11] Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Vbench++: Comprehensive and versatile benchmark suite for video generative models. *CoRR*, abs/2411.13503, 2024. 7, 8
- [12] Baoxiong Jia, Ting Lei, Song-Chun Zhu, and Siyuan Huang. Egotaskqa: Understanding human tasks in egocentric videos. In *NIPS*, 2022. 7
- [13] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *SIGOPS*, 2023. 4
- [14] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *ECCV*, 2018. 4
- [15] Chenyang Lei, Yazhou Xing, and Qifeng Chen. Blind video temporal consistency via deep video prior. *NIPS*, 2020. 4
- [16] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. 5
- [17] Yanghao Li, Tushar Nagarajan, Bo Xiong, and Kristen Grauman. Ego-exo: Transferring visual representations from third-person to first-person videos. In *CVPR*, 2021. 7
- [18] Kevin Qinghong Lin, Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Z Xu, Difei Gao, Rong-Cheng Tu, Wenzhe Zhao, Weijie Kong, et al. Egocentric video-language pretraining. In *NIPS*, 2022. 7
- [19] Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tiejong Zeng, Raymond Chan, and Ying Shan. Evalcrafter: Benchmarking and evaluating large video generation models. In *CVPR*, 2024. 4
- [20] Tomohiro Motoda, Masaki Murooka, Ryoichi Nakajo, Muhammad A. Muttaqien, Koshi Makihara, Hanbit Oh, Keisuke Shirai, Floris Erich, Ryo Hanai, and Yukiyasu Domae. Aist-bimanual manipulation, 2025. 7, 9
- [21] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 5
- [22] Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. Film: Frame interpolation for large motion. In *ECCV*, 2022. 6
- [23] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *NIPS*, 2016. 4
- [24] Modi Shi, Yuxiang Lu, Huijie Wang, Chengen Xie, and Qingwen Bu. Introducing agibot world colosseum: A large-scale manipulation platform for scalable and intelligent embodied systems. <https://opendrivelab.com/AgiBot-World/>, 2025. Blog post. 7, 9
- [25] Yale Song, Eugene Byrne, Tushar Nagarajan, Huiyu Wang, Miguel Martin, and Lorenzo Torresani. Ego4d goal-step: Toward hierarchical understanding of procedural activities. In *NIPS*, 2024. 7
- [26] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 4
- [27] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 4
- [28] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation. In *ICLRW*, 2019. 7
- [29] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Fei Wu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *CoRR*, abs/2503.20314, 2025. 3

- [30] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. In *ICLR*, 2023. [7](#)
- [31] Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, YINUO Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. In *RSS*, 2025. [7](#), [9](#)
- [32] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Dynamicrafter: Animating open-domain images with video diffusion priors. In *ECCV*, 2025. [1](#), [4](#), [5](#), [6](#)
- [33] Rui Zhang, Yaosen Chen, Yuegen Liu, Wei Wang, Xuming Wen, and Hongxia Wang. Tvg: A training-free transition video generation method with diffusion models. *CoRR*, abs/2408.13413, 2024. [5](#)