APPENDIX

VI. Detailed Results

A. Additional Policy Evaluation Results

1) Proof-of-Concept for Policy Evaluation for Push-T: We present an additional task for proof-of-concept for policy evaluation, which is push-T [51]. The desired goal of push-T is pushing a T-shape object to a target location with target pose. To simplied the task completion judgment, we use OpenCV to assess whether the T-shape match the target location in the last frame. The results are present in Table III. The rollout videos are available in project website.

TABLE III: **Policy Evaluation on Push-T** Pearson and MMRV scores (mean $\pm$ std) for Cosmos-Predict2-2B and IRASim.

Method	Pearson ↑	MMRV ↓
Cosmos-Predict2-2B	0.961 $\pm$ 0.026	0.042 $\pm$ 0.024
IRASim	0.610 $\pm$ 0.028	0.213 $\pm$ 0.030

2) Details for Benchmarking with Different Pre-trained Video Models: To better understand which video model performs best in the policy evaluation scenario, we consider additional baselines, as illustrated below.

- **IRASim** [20]: a diffusion transformer with 16 frames and a frame-level action-conditioning module within each transformer block.
- **Cosmos-Predict-1-7B** and **Cosmos-Predict-1-14B** [17]: We fine-tuned the `Cosmos-Predict1-7B-Video2World` and `Cosmos-Predict1-14B-Video2World` to perform next-frame prediction based on a single action. The action conditioning is applied to the time embeddings of the model as well as the description of their white paper.
- **Wan 2.1 LoRA** [19]: We fine-tuned the `Wan 2.1 I2V-14B-480P` using LoRA. The action conditioning is applied to the time embeddings of the model.
- **Cosmos-Predict-2-2B** [18]: We fine-tuned the `Cosmos-Predict2-2B-Video2World`. The action conditioning is applied to the time embeddings of the model as well.

The correlation graphs are presented in 9

3) Real-world Experiments: For the real-world experiments, we collected an additional 300 trajectories in our bridge setup to further narrow the domain gap between the Bridge dataset and our environment. Table VI reports policy evaluation results with and without these additional trajectories. Quantitatively, the results demonstrate that these trajectories are critical for reliable policy evaluation.

4) Policy Evaluation with Human Evaluation: Instead of relying solely on a VLM for task completion judgments, we also conducted human evaluations to determine whether each rollout was successfully completed. The results are presented in Table IV. While the VLM annotations are not fully accurate, we observed that the Pearson coefficient is affected whereas MMRV remains stable. This suggests that the VLM can still serve as a useful component for annotating rollout videos.

Furthermore, we compared VLM annotations against human annotations and found that the VLM is more accurate for shorter rollout videos, such as Lift and Can.

5) Failure Case from VLM: In the main paper, we mainly discuss the failure case for policy evaluation caused by the video. Here, we also provide the discussion about the failure caused by VLM judgment.

Despite carefully designing the prompt to enhance the reliability of automatic annotation, some failures still arise due to limitations of the vision-language model (VLM). The are two major failure cases.

First, although we instruct the VLM to ignore frames following task completion, it may still consider them during evaluation. Please refer to Figure 10.

Second, since we uniformly sample frames from the rollout video, failures that occur during skipped intervals may be missed by the VLM, leading to incorrect judgments. Please refer to Figure 11.

6) Failure cases from Video model:

B. Action-Conditional Video Prediction Results

We further provide more qualitative results for action-conditional video prediction for robomimic in Figure 12. Also, in our main paper, we present the policy evaluation on Bridge [53], [52], and present their action-conditional video prediction results in Table VII. Please refer to our project website for more results.

Furthermore, to investigate whether the video model suffers from accumulated error, we provide a quantitative comparison for video prediction. The results are presented in Table VIII. They show that video prediction using teacher forcing—where frames are generated conditioned on the ground-truth frames rather than the previously generated frames—outperforms the auto-regressive approach.

VII. Design Space of Action-conditional Video Prediction

We further conduct a series of experiments about the design space of adding action conditioning to an existing video model, including network architecture, training strategy and action representation. We use video prediction metrics to evaluate the resulting action-conditional model.

a) Architecture.: We investigate the factors that potentially affect the video prediction performance. First, we explore various approaches to incorporate action-conditioning into action-conditional video prediction training with diffusion models. Similar to [55], we can leverage cross-attention mechanism [56] in DiT blocks to integrate action information, as shown in Figure 13(a). Note that this modification doesn't affect the original network backbone, which potentially preserve the prior information learn from Cosmos.

Another option is to append the action token to the image token in channel dimension, as shown in Figure 13(b). This strategy requires modification to the first DiT blocks in Cosmos and could potentially affect the prior information that

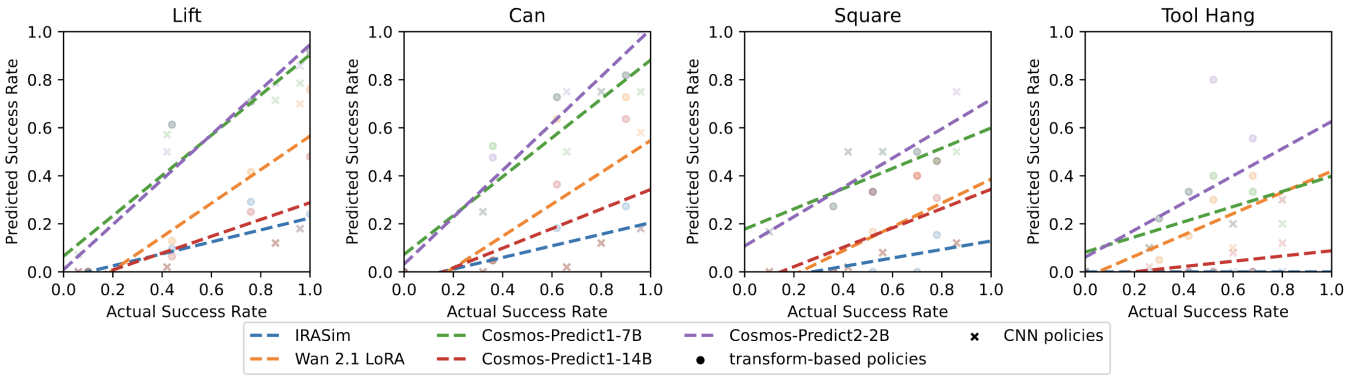


Fig. 9: **Correlation Plots for Policy Evaluation with Different Video Model**. In general, we found that Cosmos-Predict-2-2B perform the best.

TABLE IV: **Quantitative Results for Policy Evaluation in Simulation Environment with different task completion judgment**. Pearson correlation coefficient (higher is better) and MMRV (Mean Metric Rank Violation; lower is better) between video-model and simulator policy evaluations.

Task	**Lift**		**Can**		**Square**		**Tool Hang**	
	Pearson ↑	**MMRV ↓**	**Pearson ↑**	**MMRV ↓**	**Pearson ↑**	**MMRV ↓**	**Pearson ↑**	**MMRV ↓**
Cosmos-Predict-2-2B (VLM)	0.879	0.015	0.801	0.146	0.810	0.168	0.790	0.222
Cosmos-Predict-2-2B (Human Eval)	**0.888**	**0.014**	**0.860**	**0.141**	**0.849**	**0.165**	**0.833**	**0.217**

Fig. 10: Hallucination can occur after task completion. According to our prompt design, the VLM is instructed to ignore frames following the completion of the task; however, it sometimes fails to do so and includes those frames in its judgment.

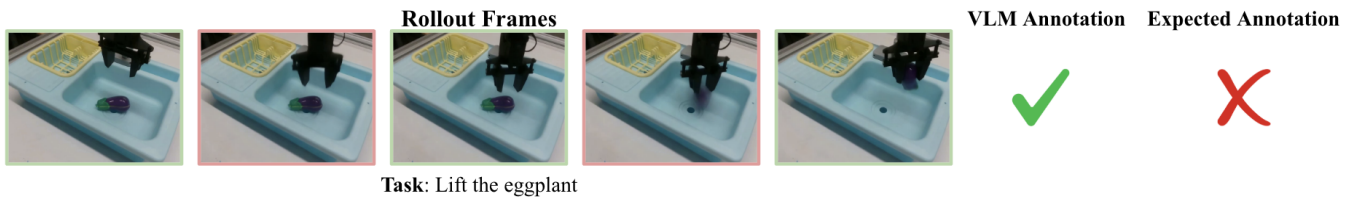


Fig. 11: Hallucination can occur in the skipped frames, so VLM skips it and incorrectly judge the rollout video as success.

TABLE V: **VLM Annotation Success Rate**. The VLM achieves higher annotation success rates on short-horizon tasks and lower rates on long-horizon tasks.

Model	**Lift**	**Can**	**Square**	**Tool Hang**
VLM Success Rate	0.79	0.71	0.68	0.66

TABLE VI: **Real-World Policy Evaluation.** Pearson and MMRV scores (mean ± std) for Cosmos and IRASim.

Method	**Pearson ↑**	**MMRV ↓**
Cosmos	0.687 ± 0.031	0.170 ± 0.024
Cosmos w/o trajectories in our bridge	0.611 ± 0.019	0.209 ± 0.022
IRASim	0.610 ± 0.028	0.213 ± 0.030

model perviously acquired. The architectures are presented in Figure 13. The results are summarized in Table IX.

They reveal that while most prior works condition diffusion models using cross-attention, we found that integrating actions through time embeddings yields the best performance.

b) Finetuned or Frozen Backbone.: In our experiments, the entire network is optimized in an end-to-end manner, which is both time-consuming and requires substantial GPU memory. An alternative approach is to freeze the backbone

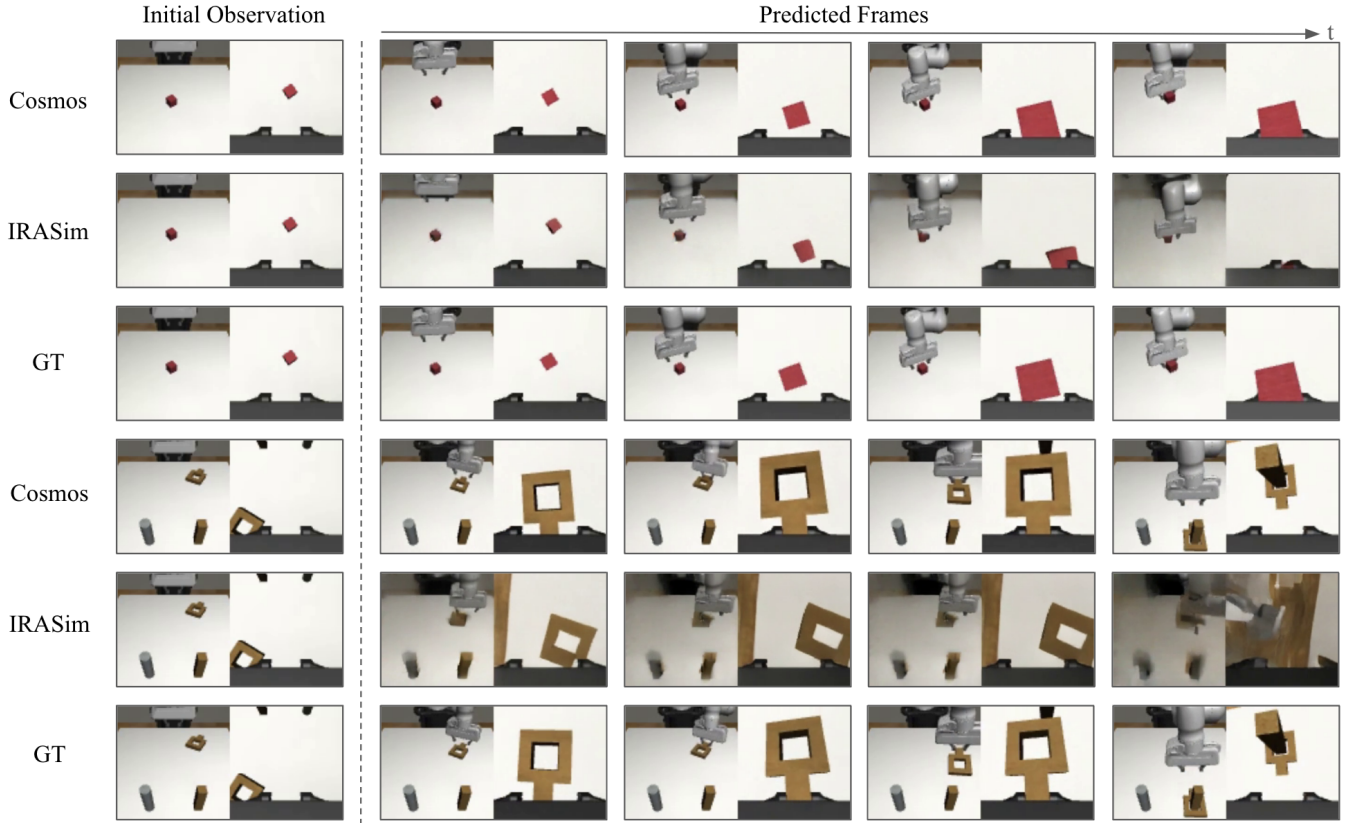


Fig. 12: **Action-conditional Video Prediction Results**. The predicted results of Cosmos are closed to the ground-truth results.

TABLE VII: Evaluation of action-based video prediction on Bridge dataset.

Method	PSNR ↑	SSIM ↑	Latent L2 ↓	FVD ↓
IRASim [20]	19.13	0.64	0.38	593
Cosmos [17]	**21.14**	**0.82**	**0.32**	**190**

and fine-tune only the action encoder, reducing GPU memory consumption and improving training efficiency. However, our experimental results indicate that this strategy leads to suboptimal performance.

c) Action Representation.: Since raw actions could be low-frequency signals, we adopt positional embedding to project them to a high-frequency representation. In Table IX, we also present the video prediction from the model without positional embedding, demonstrating poorer video prediction performance without it.

VIII. Experimental Details

A. Simulation Experimental Setup

1) Action-conditional Video Prediction Model:

a) Training and Evaluation Configuration.: Our action-conditional video model is trained for 240K steps with a batch size of 128 and a learning rate of 4e-4, utilizing 32 H100 GPUs for efficient large-scale training. For the RoboMimic [46] dataset, we split it into training, validation, and testing sets with an 8:1:1 ratio.

For video prediction evaluation on RoboMimic, given a single conditional frame, we generate 64 future frames in an auto-regressive manner and compare them with the ground-truth frames. For Bridge [53], [52], since the variance in trajectory length is much smaller than in RoboMimic, we condition on the first frame and generate frames auto-regressively until the end of the trajectory.

b) Policy Rollout Dataset.: Aside from using human demonstration from RoboMimic [47], [46], we also generate rollouts with policies for action-conditional video prediction model training. For each task, four policies are acquired with diffusion policy and RoboMimic dataset as well to generate rollout. The success rates these policies are:

- Lift: 0.04, 0.38, 0.82, 1.00
- Can: 0.00, 0.38, 0.56, 0.84
- Square: 0.36, 0.64, 0.74, 0.78
- Tool Hang: 0.26, 0.54, 0.64, 0.66

2) Policy Evaluation:

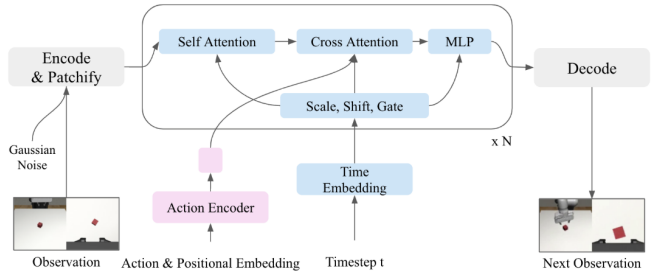
a) Policies.: We adopt Diffusion Policy [51] and RoboMimic to train a suite of policies. For each task, we provide four policies implemented with both Transformer and CNN backbones, respectively. The performance is evaluated by rolling out each policy in a simulator across 50 randomized initial configurations. The success rates for each task are as follows:

TABLE VIII: **Comparison between Auto-Regressive Generation and Teacher-Forcing in Action-Conditional Video Prediction.** We generate 64 future frames in an auto-regressive manner and evaluate the models on both RoboMimic (in-distribution) and policy rollout (out-of-distribution) trajectories.

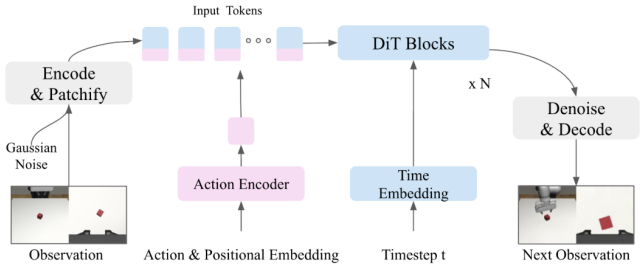
World Model	RoboMimic (In-Distribution)				Policy Rollouts (Out-of-Distribution)			
	PSNR ↑	SSIM ↑	Latent L2 ↓	FVD ↓	PSNR ↑	SSIM ↑	Latent L2 ↓	FVD ↓
Cosmos-Predict2-2B (Auto-Regressive)	20.622	0.868	0.315	98.455	15.996	0.793	0.458	250.234
Cosmos-Predict2-2B (Teacher Forcing)	**22.311**	**0.899**	**0.133**	**71.455**	**20.122**	**0.813**	**0.378**	**120.770**

TABLE IX: **Ablation Study** on making Cosmos action-conditional on the Robomimic dataset. We investigate different model architectures, training strategy and action representations.

Method	PSNR↑	SSIM↑	Latent L2↓	FVD↓
Cosmos via Time-Embedding	**20.622**	**0.868**	**0.315**	**98.455**
Cosmos via Channel-Concat	18.671	0.826	0.397	154.816
Cosmos via Cross-Attention	19.764	0.843	0.372	165.547
Cosmos w/ Frozen Backbone	13.012	0.700	0.711	661.012
Cosmos w/o Positional Embedding	20.012	0.841	0.391	128.123



(a) action-conditional via cross-attention



(b) action-conditional via channel concatenation

Fig. 13: **Design space for action-conditional next frame prediction**. Alternatively, we added action conditioning through (a) channel dimension (b) cross attention.

- **Lift**: Transformer — 0.10, 0.44, 0.76, 1.00; CNN — 0.06, 0.42, 0.80, 0.94
- **Can**: Transformer — 0.00, 0.36, 0.62, 0.90; CNN — 0.32, 0.66, 0.80, 0.96
- **Square**: Transformer — 0.26, 0.52, 0.70, 0.80; CNN — 0.10, 0.46, 0.80, 0.96
- **Tool Hang**: Transformer — 0.30, 0.42, 0.52, 0.68; CNN — 0.00, 0.26, 0.60, 0.80

B. Real-World Experimental Setup

1) Action-conditional Video Prediction Model: Our action-conditional video model is trained for 200K steps with a batch size of 128 and a learning rate of 4e-4, utilizing 32 H100 GPUs for efficient large-scale training. For Bridge, we use the dataset processed by IRASim [20].

TABLE X: **Dataset Statistics.** Each cell reports the number of trajectories / transitions for Proficient-Human (PH), Multi-Human (MH), and Policy Rollouts across different tasks.

Task	PH	MH	Policy Rollout
Lift	200 / 9k	300 / 31k	400 / 160k
Can	200 / 23k	300 / 62k	400 / 160k
Square	200 / 30k	300 / 80k	400 / 160k
Tool Hang	200 / 95k	– / –	400 / 280k

C. Policy Evaluation

a) Policies: The evaluated policies include Octo-Small, Octo-Base [3], and Open-VLA [4]. To assess real-world performance, we replicate the Bridge setup [53], [52] and execute each policy over 20 trials to measure the success rate. The results are summarized in Table XI.

TABLE XI: Performance comparison across different objects and models.

Model	Eggplant	Pot	Carrot	Cup
OpenVLA	0.30	0.50	0.50	0.70
Octo_Small	0.10	0.30	0.15	0.20
Octo_Base	0.15	0.35	0.10	0.30

IX. Auto-Annotation for Rollout Video with VLM

A. Prompt Design

To reduce the human effort required to annotate task completion rollout videos—generated by video models and

policies—we leverage Gemini-2.5-Pro [22] to automate the annotation process.

The prompt design used for Gemini-2.5-Pro comprises the following components:

- **system_prompt**: Specifies the assistant's behavior, role, or persona.
- **begin_prompt**: Marks the start of the interaction and provides the user input or task description.
- **end_prompt**: Signals the end of the input and prompts the model to generate a response.

The complete API query to Gemini-2.5-Pro consists of the system prompt, begin prompt, the set of image frames, and the end prompt. Below, we provide the specific prompts used for each task.

1) Prompts for Simulation Experiments: In our simulation experiments, there are four different tasks. The system prompt and end prompt for these tasks are the same, which are shown below:

```
system_prompt = "You are an expert in judging
     whether the robot arm performs the task
    successfully. The demonstration video
    contains two views: one is the robot arm'
    s view, and the other is the human's view
    . You are responsible for providing a
    task completion judgement, given image
    frames uniformly sampled from the
    demonstration video."

end_prompt = "Please judge whether the robot
    arm performs the task correctly with a
    binary label (Yes or No)."
```

Here are the begin prompt for each task.

Lift.

```
begin_prompt = "
Please judge the robot arm performs the task
    correctly with binary label (Yes or No)
    with success criteria and failure
    criteria. Task is successful if it
    satisfies all the success criteria and
    fails if it satisfies any of the failure
    criteria.

success criteria:
1. The cube is successfully lifted by the
    robot arm, which means the gripper is
    closed around the cube and the cube is
    moved upwards.
2. Once the task is completed, please ignore
    the rest of the video.

failure criteria:
1. The information from the human's view is
    not consistent with the robot arm's view.
2. The video is not clear and of good quality
    . For instance, the boundary of the
    object or robot arm are not sharp and
    clear.
3. The cube is not always visible in the
    video.
4. Shot changes. For instance viewpoint and
    the background should be consistent
    between frames.
5. There is distortion or blurring, making it
     difficult to see the task.
6. There is not a red cube in the video.
7. The task is not complete.
"
```

Can.

```
begin_prompt = "
Please judge the robot arm performs the task
    correctly with binary label (Yes or No)
    with success criteria and failure
    criteria. Task is successful if it
    satisfies all the success criteria and
    fails if it satisfies any of the failure
    criteria.

success criteria:
1. The arm closes around the coke can and
    moves it consistently with the robot arm'
    s motion.
2. The coke can is lifted and moved the bin.
3. Once the task is completed, please ignore
    the rest of the video.

failure criteria:
1. The information from the human's view is
    not consistent with the robot arm's view.
2. The video is not clear and of good quality
    . For instance, the boundary of the
    object or robot arm are not sharp and
    clear.
3. The coke can is not always visible in the
    video.
4. Shot changes. For instance viewpoint and
    the background should be consistent
    between frames.
5. There is distortion or blurring, making it
     difficult to see the task.
6. There is not acoke can in the video.
7. The task is not complete.
"
```

Square.

```
begin_prompt = "
Please judge the robot arm performs the task
    correctly with binary label (Yes or No)
    with success criteria and failure
    criteria. Task is successful if it
    satisfies all the success criteria and
    fails if it satisfies any of the failure
    criteria.

success criteria:
1. The square nut is successfully lifted by
    the robot arm, which means the gripper is
     closed around the square and the square
    is moved upwards.
2. The square nut is moved with the movement
    of the end-effector of the robot arm.
3. The square nut is placed onto a rod in the
     end.
4. Once the task is completed, please ignore
    the rest of the video.

failure criteria:
1. The robot arm does not complete the task.
2. The information from the human's view is
    not consistent with the robot arm's view.
```

```
3. The video is not clear and of good quality
    . For instance, the boundary of the
    square nut or robot arm are not sharp and
     clear.
4. The square nut or rod are not always
    visible in the video.
5. Shot changes. For instance viewpoint and
    the background should be consistent
    between frames.
6. The square nut or rod are not visible in
    the video.
7. The square nut or rod is moved when the
    arm has no contact with it.
"
```

Tool Hang.

```
begin_prompt = "
Please judge the robot arm performs the task
    correctly with binary label (Yes or No)
    with success criteria and failure
    criteria. Task is successful if it
    satisfies all the success criteria and
    fails if it satisfies any of the failure
    criteria.

success criteria:
1. The robot arm must assemble a frame by
    inserting a hook piece into a base.
2. The arm hang the wrench on the hook.
3. Once the task is completed, please ignore
    the rest of the video.

failure criteria:
1. The robot arm does not complete the task.
2. The the information from the human's view
    is not consistent with the robot arm's
    view.
3. The video is not clear or the quality is
    poor.
4. Shot changes. For instance viewpoint and
    the background should be consistent
    between frames.
5. The hook or wrench are not visible in the
    video.
6. The wrench is moved when the arm has no
    contact with it.
"
```

2) Prompts for Real-World Experiments: In our real-world experiments, there are also four different tasks. The system prompt and end prompt for these tasks are the same, which are shown below:

```
system_prompt = "You are an expert in judging
     whether the robot arm performs the task
    successfully. The demonstration video
    contains a black robot arm with gripper
    and a set of toy objects. You are
    responsible for providing a task
    completion judgement, given image frames
    uniformly sampled from the demonstration
    video."

end_prompt = "Please judge whether the robot
    arm performs the task correctly with a
    binary label (Yes or No)."
```

Lift the eggplant

```
begin_prompt = "
Please judge the robot arm performs the task
    correctly with binary label (Yes or No)
    with success criteria and failure
    criteria. Task is successful if it
    satisfies all the success criteria and
    fails if it satisfies any of the failure
    criteria.

success criteria:
1. The gripper closes around the eggplant.
2. The eggplant is lifted by the gripper.
3. Once the task is completed, please ignore
    the rest of the video.

failure criteria:
1. The robot arm fails to complete the task.
2. The video is unclear or of poor quality.
3. Shot changes are present. For example,
    inconsistencies in viewpoint or
    background between frames.
4. The object moves without any contact from
    the arm or gripper.
5. The object's appearance (e.g., shape or
    color) changes during the video.
7. The eggplant turns into carrot.
"
```

Lift the pot

```
begin_prompt = "
Please judge the robot arm performs the task
    correctly with binary label (Yes or No)
    with success criteria and failure
    criteria. Task is successful if it
    satisfies all the success criteria and
    fails if it satisfies any of the failure
    criteria.

success criteria:
1. The gripper closes around the pot.
2. The pot is lifted by the gripper.
3. Once the task is completed, please ignore
    the rest of the video.

failure criteria:
1. The robot arm fails to complete the task.
2. The video is unclear or of poor quality.
3. Shot changes are present. For example,
    inconsistencies in viewpoint or
    background between frames.
4. The object moves without any contact from
    the arm or gripper.
5. The object's appearance (e.g., shape or
    color) changes during the video.
6. The size of the object changes.
"
```

Lift the carrot

```
begin_prompt = "
Please judge the robot arm performs the task
    correctly with binary label (Yes or No)
    with success criteria and failure
    criteria. Task is successful if it
    satisfies all the success criteria and
    fails if it satisfies any of the failure
    criteria.
```

```
success criteria:
1. The gripper closes around the carrot.
2. The carrot is lifted by the gripper.
3. Once the task is completed, please ignore
    the rest of the video.

failure criteria:
1. The robot arm fails to complete the task.
2. The video is unclear or of poor quality.
3. Shot changes are present. For example,
    inconsistencies in viewpoint or
    background between frames.
4. The object moves without any contact from
    the arm or gripper.
5. The object's appearance (e.g., shape or
    color) changes during the video.
6. The size of the object changes.
7. The carrot turns into eggplant.
"
```

Lift the cup

```
begin_prompt = "
Please judge the robot arm performs the task
    correctly with binary label (Yes or No)
    with success criteria and failure
    criteria. Task is successful if it
    satisfies all the success criteria and
    fails if it satisfies any of the failure
    criteria.

success criteria:
1. The gripper closes around the pink cup.
2. The pink cup is lifted by the gripper.
3. Once the task is completed, please ignore
    the rest of the video.

failure criteria:
1. The robot arm fails to complete the task.
2. The video is unclear or of poor quality.
3. Shot changes are present. For example,
    inconsistencies in viewpoint or
    background between frames.
4. The object moves without any contact from
    the arm or gripper.
5. The object's appearance (e.g., shape or
    color) changes during the video.
6. The handle of the cup disappear.
"
```