Appendix

A. Extended Experiments

A.1. Experiments on Training Pipeline

Image Quality Assessment We evaluated the quality of the generated images using three metrics at Table 1: PSNR [1] (Peak Signal-to-Noise Ratio), SSIM [3] (Structural Similarity Index), and LPIPS [4] (Learned Perceptual Image Patch Similarity). Table 1 shows that our context-shared multimodal learning pipeline enables effective action generation during joint training while simultaneously improving the quality of future image prediction.

Table 1. **Image generation quality**. We evaluated three image-generation metrics using model weights from Stage 1 and Stage 2 in **unseen** experiments for 1000 attempts. Stage 1 is trained solely on image generation objectives, while Stage 2 performs joint training on both image and action generation after Stage 1.

MM-ACT	PSNR ↑	SSIM ↑	LPIPS ↓
Stage 1	12.08	0.79	0.11
Stage 2	**14.23**	**0.80**	**0.09**

Text Quality Assessment We conduct a detailed analysis and additional experiments on the generation quality of the model's text modalities.

For text modality, we compare the task planning accuracy of MM-ACT (+Text) at Stage 1 and Stage 2 on clustered unseen scenes. The evaluation dataset is similarly collected through our RoboTwin data pipeline, with the key distinction that we select unseen scenes and object spatial arrangements from the training dataset. To evaluate the correctness between our model's output and the ground truth, we leverage GPT-4o [2] with the prompt shown in Figure 2. We conduct evaluations for 1,000 attempts, and the results are presented in Table 2. In Figure 6, we further present several representative comparisons between the task planning outputs generated by our model and the corresponding ground truth annotations.

This indicates that our model acquires strong task planning capabilities during the initial Stage 1, where the text modality is trained independently. However, after training with action modality in Stage 2, the text generation performance deteriorates, which is inconsistent with the results observed in the image generation modality. We visualize the training curves of the text and image modalities during Stage 1 and Stage 2 in Figure 1.

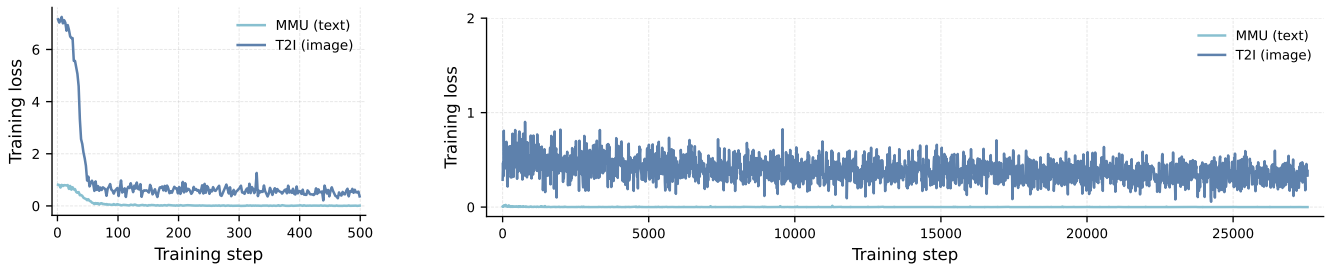
Table 2. **Text generation quality.** Stage 1 is trained solely on text-generation objectives, while Stage 2 performs joint training on both text and action generation after Stage 1. Accuracy is defined as the proportion of evaluations in which LLM judge outputs "yes" among all evaluations.

MM-ACT	Acc (%)
Stage 1	**81.5**
Stage 2	68.7

As observed, the training loss for text modality during Stage 1 rapidly approaches 0 in about 100 steps, indicating a near-perfect fitting in our dataset. In contrast, the loss for image modality continues to decrease consistently throughout both Stage 1 and Stage 2. This suggests that text modality is prone to overfitting with increased training steps, resulting in decreased generalization performance on unseen scenarios. Meanwhile, the slower fitting process of the image modality allows it to continuously benefit from our training pipeline, achieving steady improvements.

A.2. Ablation Study

Action Decoding Strategy Different from the re-mask decoding strategy employed for image and text modalities, we adopt a one-step parallel decoding strategy for action prediction, significantly accelerating the speed and reducing the frequency of model forward processes. To comprehensively analyze the effectiveness and computational efficiency of these two decoding strategies, we conduct comparative experiments under two settings: action chunk sizes of 8 and 16. The experimental results are summarized as Table 3: As illustrated in table, when action chunk size is set to 8, the re-mask decoding strategy does not enhance action generation performance but instead leads to nearly a five-fold increase in inference time. However, when the action chunk size is increased to 16, incorporating the re-mask strategy indeed improves action generation success rate, albeit still accompanied by a significant rise in inference duration. This indicates that the re-mask parallel de-



(a) Stage1 (b) Stage2

Figure 1. **Comparison of training loss between MMU (text) and T2I (image) on two training stages.** In Stage 1, the loss of text modality rapidly converges to a value very close to 0, while the loss of image modality quickly decreases below 1 and then declines slowly. In Stage 2, the loss of text modality remains consistently near 0, whereas image modality loss exhibits a slow, oscillating decline over an extended number of training steps.

```
You are a judge for embodied task
planning.  Your job:
- Compare an agent's plan (agent_plan)
with a reference plan (ground_truth).
- Decide whether they are consistent in
terms of task planning and decisions for
the current task.
- The two plans do NOT need to be exactly
the same; similar intent and decision
logic are enough.
Requirements:
- If they are consistent, answer exactly
``yes''.
- If they are not consistent, answer
exactly ``no''.
- Do not output anything else.
Here is the data:
agent_plan:  {agent_plan}.
ground_truth:  {ground_truth}
```

Figure 2. **Prompt used for LLM judge in our experiments.** {agent_plan} is our model's output, {ground_truth} denotes the task planning annotation corresponding to the evaluation sample.

Table 3. **Action decoding strategy**. "one-step PD" denotes that the model generates actions using a one-step parallel decoding strategy. "Re-mask PD" indicates that the model performs multiple forward processes with iterative re-masking. "cs" refers to the chunk size, and "t" denotes the number of forward steps.

MM-ACT	Overall Avg (%)	Time
one step PD, cs=8, t=1	**43.13**	0.22s
re-mask PD, cs=8, t=6	42.38(-0.75%)	1.06s
one step PD, cs=16, t=1	43.75	0.23s
re-mask PD, cs=16, t=6	**56.75**(+13.00%)	1.06s

coding strategy yields more pronounced improvements in generation quality when applied to tasks requiring longer parallel token sequences.

Considering the real-time control requirements in robotic tasks, we ultimately select the one-step parallel decoding strategy with an action chunk size of 8 for action generation, enabling a high generation frequency of up to 40 Hz (5Hz per action chunk).

Text and Image Decoding Strategies The ablation study on decoding strategies of text and image modalities is presented in Table 4, demonstrates that during action generation training, both one-step parallel decoding and re-mask decoding strategies for text and image modalities can provide beneficial improvements to action generation performance. Besides, employing the re-mask decoding strategy in the image modality provides greater benefits compared to the one-step decoding strategy, potentially because the re-mask decoding strategy aligns more closely with the pre-training paradigm of the base model.

Table 4. **Ablation on decoding strategy for text and image modalities in RoboTwin2.0 Tasks.**

Model	Decoding Strategy	SR (%)
MM-ACT (+Text)	re-mask	46.50(+3.37%)
MM-ACT (+Text)	one-step	**46.63**(+3.50%)
MM-ACT (+Image)	re-mask	**48.75**(+5.62%)
MM-ACT (+Image)	one-step	46.13(+3.00%)
MM-ACT (Vanilla)	-	43.13

State in Text or Image's Context The ablation study on the inclusion of robot's state in context is presented in Table 5, reveals that whether incorporating robot's state in the context of text and image modalities could both provide improvements to action generation performance, though enhancement varies across the two modalities. In text modality, including the robot's state as part of the context reduces

the beneficial effect on action generation. In contrast, in image modality, incorporating the robot's state into the context enhances the beneficial effect on action generation, possibly because image generation aligns more finely with action generation, allowing closer context to mutually reinforce performance improvements.

Table 5. **Ablation study on the inclusion of robot's state in text's or image's context in RoboTwin2.0 Tasks.**

Model	State	SR (%)
MM-ACT (+Text)	without	**46.50**(+3.37%)
MM-ACT (+Text)	with	43.50(+0.37%)
MM-ACT (+Image)	without	48.75(+5.62%)
MM-ACT (+Image)	with	**51.50**(+8.37%)
MM-ACT (Vanilla)	-	43.13

A.3. Training details

We train our model with batch size of 128 and action chunk size of 8 across three experiments. For LIBERO benchmark, all training is conducted with a batch size of 128 and a learning rate of 5×10^{-5}. Among the reported model weights, LIBERO Object is trained for about 9k steps, LIBERO Spatial for about 7.5k steps, LIBERO Goal for about 8.5k steps, and LIBERO Long for about 17.5k steps. For "+Text" in LIBERO Long, we first train the text generation for one epoch (about 800 steps), with $\lambda_{\text{mmu}} = 1$ in this stage. Subsequently, we jointly train the text and action generation modalities, assigning a weight of 0.05 to λ_{mmu} and a weight of 1 to λ_{mm2a}.

For RoboTwin benchmark, we maintain the same batch size and learning rate as described above. For Vanilla model, training begins from the base model weights and continues for approximately 27k steps (5 epochs). For both the "+Text" and "+Image" models, we initially train each modality independently for 500 steps, assigning λ_{mmu} or λ_{t2i} to 1 during this stage. We then jointly train the modality together with actions for approximately 27k steps (5 epochs), assigning a modality weight of 0.1 for either λ_{mmu} or λ_{t2i}, and with $\lambda_{\text{mm2a}} = 1$. For the "+Text&Image" model, we first jointly train the text and image modalities for 500 steps, setting the weights of both λ_{mmu} and λ_{t2i} to 1. Subsequently, we train all three modalities together, assigning a weight of 1 to λ_{mm2a}, 0.05 to both λ_{mmu} and λ_{t2i}. For baselines in RoboTwin, we train them using the same batch size, learning rate, action chunk size and gradient steps as ours. For baselines in Franka real-world experiment, we use their default fine-tuning hyperparameters and evaluate the best checkpoint selected within 30k training steps. For Franka real-world experiments, we train for roughly 8k steps per task.

B. Dataset Construction & Annotation Details

In both simulated and real-world experiments with Franka arm, we adopt the robot end-effector's delta pose as the action representation. For RoboTwin experiments, we directly use the robot end-effector's absolute pose as the action.

For text annotations in LIBERO Long dataset, we first manually compose several sub-task categories corresponding to each of the 10 tasks. Then, we manually match the sub-task annotation within its corresponding episode and annotate the specific frames indicating transitions between sub-tasks. Thus, each episode is annotated into several sections by these specific frames, with each section corresponding to specific descriptions.

The expert data from RoboTwin is annotated through a low-level, rule-based path planning procedure, which involves a fixed sequence of skill function call for each task. Building on this predefined function call sequence, we label task-relevant sub-tasks for each function call and timestamped the completion of each call during expert data generation. This approach enables us to automatically obtain expert data with corresponding task-level language annotations directly from the automated data collection pipeline. Figure 3 illustrates an example of how we construct the task planning annotation. Ultimately, we expand these annotations into structured task planning texts according to predefined templates, which were then utilized for the final training. This method ensures that each frame in our expert dataset includes textual task-planning annotations.

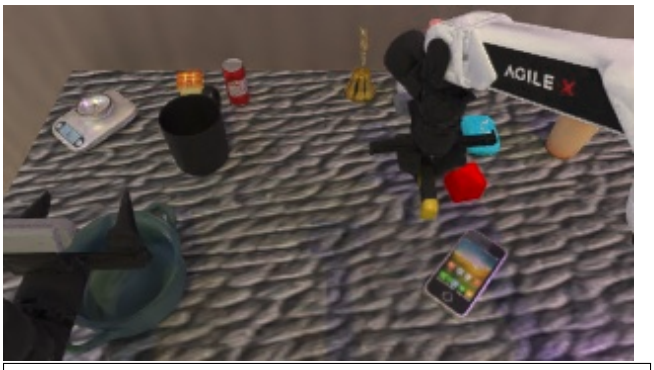
For future image prediction annotations, we directly select the frame after executing the corresponding action chunk as the ground truth for future image prediction. This method allows us to leverage the temporal nature of the original dataset without requiring additional manual annotations. Based on the aforementioned approach, we can utilize an automated data collection pipeline to gather corresponding frame-by-frame action, text, and image data from Robotwin2.0 simulation for training our model.

C. More details of Re-mask Parallel Decoding

In our framework, the re-mask strategies for text and image modalities differ in design. For text modality, the mask schedule function f_{modal} is simply set linear as: $f_{\text{modal}}(t) = t$. For image modality, we adopt cosine schedule function, set $f_{\text{modal}} = \cos\left(\frac{\pi}{2}(1-t)\right)$.

During inference, the number of masked tokens to predict at each timestep for both text and image modalities is determined according to the noise schedule function. For text and image modalities, predicted tokens with higher confidence are preferentially selected for retain, while predicted tokens with lower confidence are re-masked. In our practice, we set the temperature to 0 and do not incorporate classifier-free guidance (CFG).

ID	Item	Example Content
[1]	`instructions`	Grab the black and yellow hammer grip, then hit the block.
[2]	`planning_text`	Grasp the hammer with the right arm → Lift the hammer upwards → Move the hammer over the block → place it down to beat the block.
[3]	`history_text`	Grasp the hammer with the right arm, Lift the hammer upwards
[4]	`subtask_text`	Move the hammer over the block, place it down to beat the block.



Resulting Description:

My task is **[1]** Grab the black and yellow hammer grip, then hit the block. I need to finish this task by **[2]** Grasp the hammer with the right arm, Lift the hammer upwards, Move the hammer over the block and place it down to beat the block. Currently, I have finished **[3]** Grasp the hammer with the right arm, Lift the hammer upwards. So now I should continue to **[4]** Move the hammer over the block and place it down to beat the block.

Figure 3. **Task planning annotation example in RoboTwin2.0.** `instructions`, `planning_text`, `history_text`, and `subtask_text` are concatenated into a single annotation.

For action modality, we introduce a one-step parallel decoding strategy during both training and inference, requiring the model to predict all masked tokens within a single forward process. Regarding the re-mask decoding strategy, which we also apply to the action modality, our implementation aligns consistently with that of image modality.

D. Robotic Embodiments in Simulation & Real-World

In the LIBERO and real-world experiments, we use Franka as the embodiment; in the RoboTwin experiment, we use Aloha-AgileX as the embodiment. The specific embodiments are visualized in Figure 4. Franka Research 3 is a force-sensitive robotic system designed for robotics and artificial intelligence research. The system features a 7 degree-of-freedom (DoF) arm with integrated torque and force sensors at each joint, supporting control frequencies of up to 1 kHz. Aloha-AgileX is a robotic platform integrating a mobile base with dual manipulator arms, enabling whole-body teleoperation of both the base and the arms. The system supports synchronous coordination of the differential-drive chassis and the bimanual arms, thus expanding the operational workspace beyond static manipulators. By combining the base's linear and angular velocity control with the manipulators' multiple joint actuations (e.g., the original ALOHA system features approximately 14DoF), the platform realizes a high-dimensional action space for research in mobile manipulation, bimanual coordination, and simulation-to-real-world transfer.

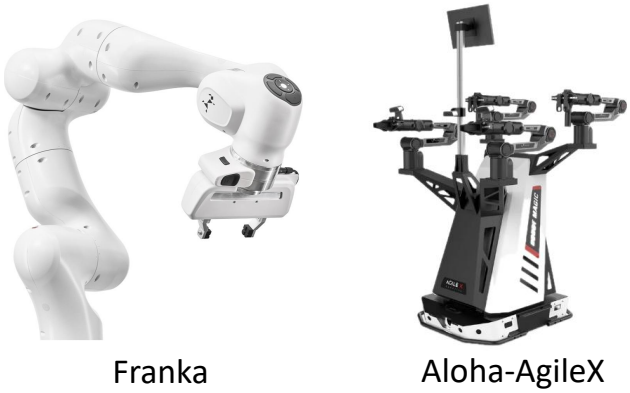


Figure 4. **Embodiments used in simulation and real-world experiments.**

E. Task Visualizations on RoboTwin2.0

In our experiments, we used eight tasks from the RoboTwin 2.0 simulation benchmark. The following are detailed descriptions of each task, along with visualizations with domain randomization, as shown in Figure 5:

- **Adjust Bottle**: Pick up the bottle on the table and place it upright using the correct arm.
- **Beat Block Hammer**: There is a hammer and a block on the table; use the arm to grab the hammer and strike the block.
- **Click Bell**: Click the top center of the bell on the table.
- **Dump Bin Bigbin**: Grab the small bin and pour the balls into the big bin.
- **Lift Pot**: Use the arm to lift the pot.
- **Move Playingcard Away**: Use the arm to pick up the playing card and move it away from the table. For example, if the playing card is on the outward side of the table, you should move it further outward.
- **Place Burger Fries**: Use both arms to pick up the burger and fries and place them onto the tray.
- **Place Can Basket**: Use one arm to pick up the can and place it into the basket, while the other arm lifts the basket.

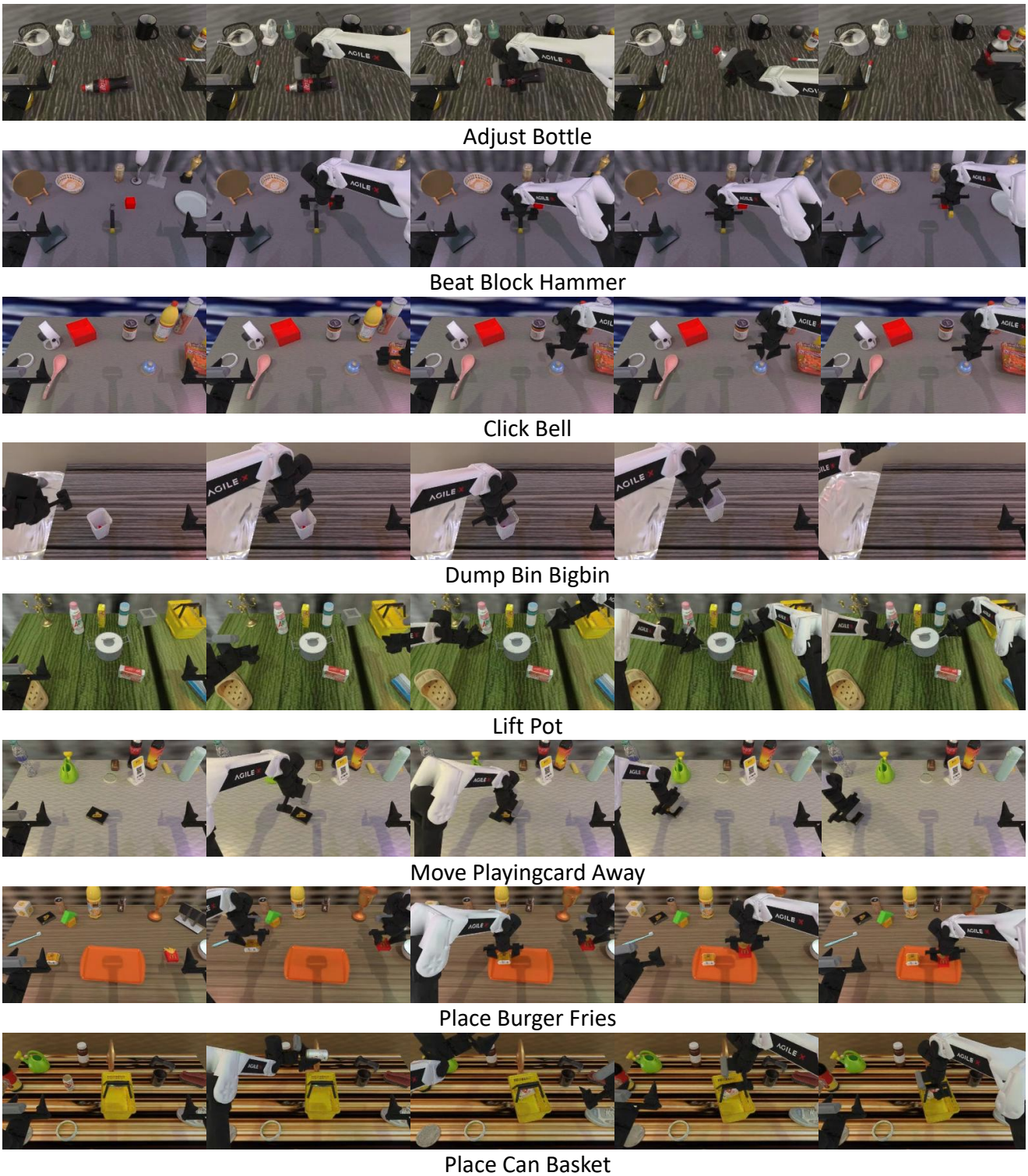


Figure 5. **Visualization of eight tasks in RoboTwin2.0.**

Task: Adjust Bottle
Generated Text: My task is Lift the Coca-Cola bottle from the table upright. I need to finish this task by Grasp the bottle with the left arm, Move the bottle to the target position. So now I should continue to Grasp the bottle with the left arm.
Ground Truth: My task is Lift the Coca-Cola bottle from the table upright. I need to finish this task by Grasp the bottle with the left arm, Move the bottle to the target position. So now I should continue to Grasp the bottle with the left arm.

Task: Move Playingcard Away
Generated Text: My task is Place the blue carton for playing-cards away from the table outward.. I need to finish this task by Grasp the with the right arm, Move the playing cards away the right arm. So now I should continue to Grasp the playing with the right arm.
Ground Truth: My task is Place the blue carton for playing-cards away from the table outward.. I need to finish this task by Grasp the playing cards with right arm, Move the playing cards away with right arm. So now I should continue to Grasp the playing cards with right arm.

Task: Dump Bin Bigbin
Generated Text: My task is Take the compact tabletop trash-bin and pour the balls out.. I need to finish this task by Grasp the trash bin with the right arm, Move the bin to the middle of the middle table and place it down, Grasp the trash bin with the left arm, and move the right arm back to the origin. Currently, and, Move the bin to the middle of the middle table and place it down. So now I should continue to Grasp the trash bin with the right arm.
Ground Truth: My task is Take the compact tabletop trashbin and pour the balls out.. I need to finish this task by Grasp the trash bin with the right arm, Move the trash bin to the middle of the table and place it down, Grasp the trash bin with the left arm, and move the right arm back to the origin, Move the trash bin over the big dustbin, and shake it to dump the garbage inside. So now I should continue to Grasp the trash bin with the right arm.

Task: Beat Block Hammer
Generated Text: My task is Use the right arm to grab the hammer with claw-shaped end and beat block. I need to finish this task by Grasp the hammer with the right arm, Move the hammer to the target position, Lift the hammer with the arms upward. Currently, I have finished Grasp the hammer with the right arm, Move the hammer to the target position. So I should continue to Lift the hammer with the arms down.
Ground Truth: My task is Use the right arm to grab the hammer with claw-shaped end and beat block. I need to finish this task by Grasp the hammer with the right arm, Lift the hammer upwards, Move the hammer over the block and place it down to beat the block. Currently, I have finished Grasp the hammer with the right arm, Lift the hammer upwards. So now I should continue to Move the hammer over the block and place it down to beat the block.

Figure 6. **Visualization of text generation by MM-ACT on RoboTwin environments.** In each example, the first part is the task name, the second part is the text generated by our model, and the third part is the ground truth.

References

[1] Rafael C Gonzalez and Richard E Woods. *Digital Image Processing*. Prentice Hall, 2002. 1

[2] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 1

[3] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 1

[4] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 1