LLM-Guided Future Hypotheses for Horizon-Aware Exploration in Multi-Step Robot Manipulation

Mohammad Khoshnazar[1], Andrew Melnik[1], Michael Beetz[1]

***Abstract*— Multi-step robot manipulation requires acting under uncertainty about how the scene will evolve, making exploration and policy adaptation challenging. We study whether short-horizon, task-consistent future videos can provide useful structured priors for control and reinforcement-learning fine-tuning.**

We formalize this idea through Future-Experience Conditioning (FEC), a simple interface that conditions closed-loop policies on a latent representation of a short future video. In our simulation setup, future clips are generated in three stages, an LLM reasoner operating over a task ontology initialized from the current scene state, a robot-free digital-twin rollout of the intended object motion, and a mask-free video diffusion model that synthesizes a robot-consistent future clip without requiring segmentation at inference.

We instantiate this future-conditioning interface primarily with BC and BC+RL, and compare against a future-conditioned Streaming Flow Policy (SFP) [1] baseline on RoboCasa and CALVIN under NoFuture, GTFuture, GenFuture, and WrongFuture. Generated futures improve performance over no-future conditioning, while mismatched futures degrade it, and our BC+RL instantiation achieves the strongest overall results. An average BC+RL learning-curve analysis across 8 CALVIN tasks further shows that GTFuture improves fastest, GenFuture improves earlier and to a higher level than NoFuture, and WrongFuture remains at zero throughout training. These results suggest that short-horizon future videos can serve as useful structured priors for exploration and policy adaptation under imperfect future predictions. `https://enact2026.github.io/`

I. Introduction

Many manipulation tasks require more than reactive control. To open a drawer, move a light switch, or push an object into a drawer, a robot must account for how present actions affect future interaction states over multiple steps. Success depends on delayed consequences such as contact timing, articulation response, and task progress. These challenges are broadly reflected in recent surveys of deep reinforcement learning and contact-rich robotic manipulation [2], [3].

Recent generative video models make it possible to use a short visual hypothesis of how the scene and manipulated object should evolve. If informative, such a future can improve contact timing, disambiguate intent, and help recovery from drift. The challenge is that the future signal used during training differs from the generated frames available at test time and may be noisy or misaligned temporally. More broadly, this question sits at the intersection of visual foresight, latent world models, and large-scale robot policy learning [4]–[10].

[1]Institute of Artificial Intelligence, University of Bremen, Germany.

In this work, we study short-horizon future hypotheses as a mechanism for horizon-aware exploration and policy adaptation in multi-step manipulation. We formalize this idea with **Future-Experience Conditioning (FEC)**, which conditions a closed-loop policy on a latent representation of a short-horizon future clip. Our framework takes a task command and current scene state, initializes a task ontology, and uses an **LLM reasoner** operating over this ontology to infer what object should be manipulated, which interaction part is relevant, and how the object state should change. It then produces a robot-free digital-twin rollout for the intended object motion and uses a mask-free video diffusion model to inpaint the robot into this rollout as a future video. This video is encoded into a future latent and provided, together with recent observations, to the controller for action generation (Fig. 1).

Generated future clips provide a form of *horizon-aware exploration*, allowing the controller to condition on plausible near-future task evolution rather than acting blindly. By comparing correct and mismatched future hypotheses, we study when such structured priors help or hinder closed-loop manipulation.

This paper focuses on the role of future-conditioned visual priors in *simulation*. Our goal is not to claim real-world transfer, end-to-end digital-twin construction from raw sensory input alone, or continuous online twin updates during closed-loop execution. Instead, we study whether generated short-horizon future videos can improve policy execution and reinforcement-learning-based adaptation under controlled train and test future mismatch.

Our main controller is **BC+RL**, while a future-conditioned **SFP** model is trained as a comparison baseline under the same future-conditioning interface.

A. Contributions

- **Future hypotheses as structured priors:** We study short-horizon, task-consistent future videos as conditioning signals for multi-step robot manipulation and reinforcement-learning fine-tuning.
- **Future-Experience Conditioning (FEC):** We introduce a simple future-conditioning interface that compresses short-horizon future videos into temporally binned latent representations for closed-loop policy conditioning.
- **Empirical study with BC+RL as the main instantiation:** On RoboCasa and CALVIN, we evaluate BC, BC+RL, and a future-conditioned SFP comparison baseline under NoFuture, GTFuture, GenFuture, and

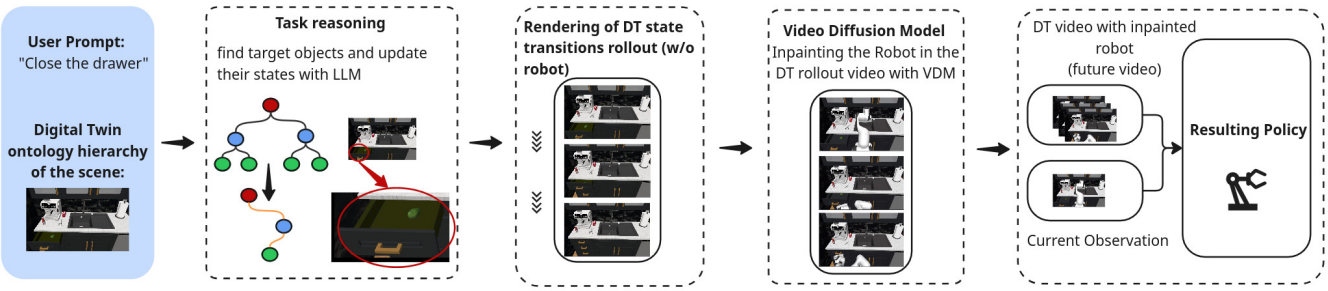

Fig. 1. Overview of our **Future-Experience Conditioning (FEC)** pipeline. From the task prompt and current scene state, an ontology-guided LLM infers the target object, interaction part, desired state transition, and rollout constraints. This drives a robot-free digital-twin rollout ("transparent" video). A mask-free video diffusion model then inpaints the robot to generate a task-consistent future clip. The policy conditions on future latents together with recent observations.

WrongFuture. Our BC+RL instantiation is strongest overall, and an average BC+RL learning-curve analysis across 8 CALVIN tasks shows that informative generated futures can improve learning speed during RL fine-tuning while mismatched futures hinder adaptation.

II. Related Work

Exploration is central to multi-step decision making in reinforcement learning and robotics, especially in manipulation where success depends on discovering interaction-relevant contacts and delayed consequences. Recent surveys review deep RL for robotic manipulation and contact-rich interaction settings, as well as broader reinforcement-learning applications across robotics domains [2], [3], [11]. Our work is most closely related to approaches that guide behavior with structured predictions of future task evolution.

Visual anticipation methods predict future observations for planning and control [4], while latent world models learn compact dynamics for longer-horizon behavior [5]–[7]. Sequence-modeling and generative control perspectives such as Decision Transformer and Diffuser are also closely related to the broader idea of exploiting predicted future structure for action selection [12], [13]. More recent robotics approaches use generated futures either as intermediate visual goals [14] or in closed-loop replanning pipelines [15]. Our setting differs in focusing on *imperfect future supervision*, where the future signal used during training differs from the generated future available at test time.

For future generation, we build on diffusion-based video modeling and inpainting [16]–[18], specifically CogVideoX [19] and VideoPainter [20], to construct a mask-free VDM conditioned on a robot-free digital-twin rollout. Our work is also adjacent to recent vision-language and digital-twin style manipulation systems that use high-level reasoning or imagined goal states for rearrangement and control [21], [22].

For robot control, our main controller is a behavior-cloning pipeline followed by reinforcement-learning fine-tuning (BC+RL), together with a BC-only variant. We additionally adapt Streaming Flow Policy (SFP) [1] to the same future-conditioning interface as a comparison baseline. More broadly, the paper is related to recent visuomotor and multi-modal robot policies including Diffusion Policy, RT-1, RT-2, CLIPort, PerAct, VIMA, and Octo [8]–[10], [23]–[26]. Our formulation is also related to prior work on imitation learning and train–test mismatch, including behavior transformers, DAgger, and scheduled sampling [27]–[29].

III. Methods

A. Problem Formulation

We consider multi-step manipulation in interactive kitchen environments. At each control step t, the robot receives observations o_t and performs a continuous action consisting of an end-effector pose delta and a gripper command. Successful behavior depends on predicting near-future transitions such as contact and articulation. We therefore study whether short-horizon future hypotheses can provide a useful structured prior for decision making and reinforcement-learning-based policy adaptation.

B. Future-Experience Conditioning (FEC)

FEC conditions a reactive controller on recent observations and a latent representation of a short-horizon future. During training, the future segment is taken from demonstrations, whereas at test time it is provided by our video generation pipeline.

a) Per-frame future embeddings: Let $v^{\text{fut}}_{1:T_f}$ denote the short-horizon future video segment available in a training sample, with T_f future frames. A frame-wise encoder $E(\cdot)$ produces per-frame embeddings:

$$e_{1:T_f} = E\left(v^{\text{fut}}_{1:T_f}\right), \qquad e_i \in \mathbb{R}^{d_e}. \tag{1}$$

b) Temporal binning and projection: We compress the future embeddings $e_{1:T_f}$ into B temporal bins using adaptive average pooling along time:

$$u_{1:B} = \text{Pool}_B\left(e_{1:T_f}\right), \qquad u_b \in \mathbb{R}^{d_e}. \tag{2}$$

Concatenating the bin vectors and applying a learned projection produces a fixed-dimensional future-conditioning vector:

$$g = P([u_1; \ldots; u_B]), \qquad g \in \mathbb{R}^{d_g}. \tag{3}$$

This g is used to condition the policy during training and inference. Temporal misalignment is expected to matter because future conditioning is computed from frame-wise embeddings before binning and projection: shifting the future sequence changes which embeddings fall into each temporal bin and therefore changes the resulting future latent.

c) Temporal-shift protocol: A natural robustness test in this formulation is to offset the future sequence by a small number of frames *before* temporal binning and projection, then recompute the future-conditioning vector from the shifted frame-wise embeddings. This is the appropriate way to test temporal misalignment in our interface, since perturbing the final pooled latent would mix temporal corruption with representation corruption. Because the controller operates on temporally pooled future embeddings rather than raw frames, mild offsets are expected to be less disruptive than larger ones. In particular, with $T = 16$ and $B = 4$, a shift of about two frames can still preserve most semantic content within adjacent bins, whereas larger offsets change bin assignments more substantially.

d) Train and test future source: Our framework trains the policy with a future experience signal in addition to an observation history, but its source differs between training and inference. During **inference**, ground-truth future frames are unavailable, so we replace them with **generated** future clips produced by the DT rollout followed by mask-free diffusion inpainting. Our experiments report controlled conditions that replace the future source at test time (ground-truth vs. generated) to quantify the train–test gap.

C. Framework Instantiation of FEC

Our framework instantiates FEC by constructing a future clip $v_{1:T}$ via an LLM-guided task-grounding step, a robot-free DT articulation rollout, and robot inpainting by a diffusion model.

1) Task Grounding and DT Goal Determination: Given the current scene state and a natural-language command c, our framework initializes a task ontology for grounding the manipulation goal. An LLM reasoner operating over this ontology infers a compact structured task specification, including the manipulated object, relevant interaction part, desired state transition, and rollout constraints. This structured output is then deterministically mapped to the DT articulation rollout and does not directly generate low-level robot actions. In all experiments, we use **GPT-4o** as the back-end reasoner. The scene state used for task grounding is obtained from the simulator at task initialization and is used only for ontology initialization and DT rollout generation. In the current implementation, GPT-4o is used in a structured JSON pipeline rather than free-form text generation. Concretely, it is used to enrich and verify ontology entries from rendered instance images, to fill the task-grounding schema from the command and initialized scene state, and optionally to validate task completion from a screenshot. For planning and verification we use deterministic decoding, while ontology enrichment uses a small nonzero temperature. When verification fails to parse, the system falls back to the sanitized ontology produced by the rule-based lint pass.

2) Robot-Free Digital Twin Articulation Video: From the ontology-grounded task specification corresponding to command c, a DT rollout generates a transparent video $\tilde{v}_{1:T}$ showing only the environment and target object motion from target camera viewpoints. This specifies the intended object evolution without requiring the DT to synthesize robot kinematics or contacts.

3) Robot Inpainting with Video Diffusion: Given $\tilde{v}_{1:T}$, a video diffusion inpainting model generates a robot-containing future clip $v_{1:T}$ that is visually consistent with the DT dynamics and camera angle. Our inpainting model is built on the CogVideoX diffusion backbone [19], using the VideoPainter framework [20] as a starting point. We modify the pipeline to be mask-free and robot-conditioned: the model receives only an initial image containing the robot and the DT "transparent" video, and outputs a robot-inpainted future clip. Training uses paired clips, with robot-free DT video as conditioning and the corresponding robot-execution clip as target reconstruction. We condition on the first frame to anchor robot appearance and initial pose.

a) Mask-free 32-channel branch: To avoid requiring privileged robot masks at inference, we train a mask-free variant based on CogVideoX [19] and our adaptation of VideoPainter [20]. The inpainting branch uses a 32-channel configuration. Thus, *no robot masks or segmentations are provided at inference*, and the model must infer robot pixels and motion from DT dynamics plus the initial frame anchor. The architecture of this mask-free diffusion model is shown in Fig. 2.

D. BC and BC with RL Fine-Tuning

Our main controller instantiation is a behavior-cloning pipeline followed by reinforcement-learning fine-tuning. The BC policy maps the recent observation history and future-conditioning vector to an action,

$$a_t^{\mathrm{BC}} = \pi_\phi^{\mathrm{BC}}(o_{t-K:t}, g_t, c), \tag{4}$$

which defines the **BC-only** baseline.

We then initialize the policy from the BC solution and further optimize it with reinforcement learning:

$$a_t = a_t^{\mathrm{BC}} + \Delta a_t, \tag{5}$$

where Δa_t denotes the policy change induced by RL fine-tuning from the BC initialization. When future conditioning is enabled, the RL-fine-tuned policy uses the same future-conditioning vector g_t as the BC initialization.

E. Future-Conditioned Streaming Flow Policy Baseline

We also adapt a Streaming Flow Policy (SFP) [1] as a comparison baseline, conditioning it on both the recent past and the same future signal used by our BC and BC+RL controllers. At time t, the policy conditions on a short observation history $o_{t-K:t}$ and a future-conditioning vector g:

$$a_t \sim \pi_\theta(\cdot \mid o_{t-K:t},\, g,\, c)\,. \tag{6}$$

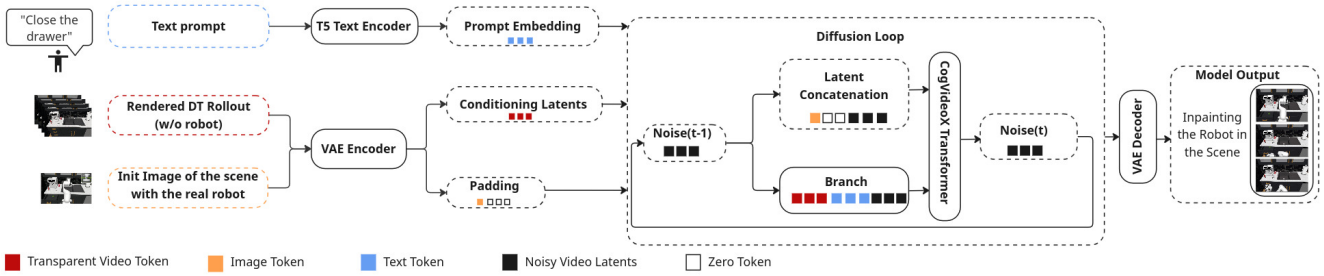

Fig. 2. Mask-free diffusion model used for robot inpainting. Conditioning inputs include an initial image with the robot, a text prompt, and the DT transparent video. The model uses a CogVideoX backbone [19] with a VideoPainter-style conditioning branch [20].

F. Training Objective

We train SFP using a conditional flow-matching objective over short action chunks. For a sampled action chunk $\sigma \in \mathbb{R}^{T_{\text{train}} \times d_a}$ and its finite-difference velocity $\dot{\sigma}$, we sample a continuous time $\lambda \sim \mathcal{U}[0,1]$ and noise $\epsilon \sim \mathcal{N}(0, I)$, and define:

$$\eta(\lambda) = \sigma_0 e^{-k\lambda}\epsilon, \qquad a = \sigma + \eta(\lambda), \qquad v = \dot{\sigma} - k\,\eta(\lambda). \tag{7}$$

The network predicts a velocity field $v_\theta(a, \lambda \mid o_{t-K:t}, g)$ and is trained with an MSE loss:

$$\min_\theta \; \mathbb{E}\Big[\big\|v_\theta(a, \lambda \mid o_{t-K:t}, g) - v\big\|_2^2\Big]. \tag{8}$$

G. Training Procedure

We train the VDM on paired DT/robot video clips using fixed-length windows and first-frame anchoring. Our main controller pipeline is based on behavior cloning followed by RL fine-tuning. We first train a BC-only controller by supervised imitation on demonstrations, then continue optimization with RL fine-tuning in the simulator to obtain BC+RL. At inference time, we replace demonstration futures with generated futures from our framework to match deployment. For comparison, we also train an SFP baseline using the same future-conditioning interface.

IV. Experiments

A. Experimental Setup

We emphasize that these experiments are a simulation study of the future-conditioning interface and its effect on policy execution and RL fine-tuning; they are not intended as a full evaluation of digital-twin fidelity, raw-perception scene understanding, or sim-to-real transfer.

Benchmarks and tasks. We evaluate our framework on **RoboCasa** and **CALVIN**. On **RoboCasa**, we evaluate **CloseDrawer**. On **CALVIN**, we evaluate `open_drawer`, `close_drawer`, `turn_on_lightbulb`, `turn_off_lightbulb`, `turn_on_led`, `turn_off_led`, `push_into_drawer`, and `move_slider_left`.

Exploration viewpoint. In BC+RL, the four future conditions also define four exploration regimes. **NoFuture** removes explicit future guidance, **GTFuture** provides an oracle short-horizon prior, **GenFuture** provides the realistic imperfect prior available at deployment, and **WrongFuture** provides an intentionally misleading prior. This lets us test whether future clips help RL fine-tuning because they provide structured exploration guidance rather than because they simply add extra input.

Scope of grounding. The task-grounding module is used only once at task initialization to create the ontology-grounded DT rollout. During closed-loop execution, the controller does not repeatedly query the LLM reasoner or update the ontology online; it acts from recent observations together with the future-conditioning vector.

Implementation details. In all experiments, the future clip contains $T = 16$ frames. We compress per-frame future embeddings into $B = 4$ temporal bins and project them to a latent of dimension $d_g = 256$. For the BC/BC+RL controllers, future frames are encoded frame-wise with a ResNet-18 backbone pretrained on ImageNet before temporal binning and projection. Our main RL experiments use BC-initialized TD3-style fine-tuning with batch size 256, discount 0.99, target update $\tau = 0.005$, policy delay 2, actor and critic learning rates 5×10^{-5}, policy noise 0.05, noise clip 0.08, and 140k environment steps. For task grounding, GPT-4o is used once at task initialization to fill the structured ontology-based schema that specifies the manipulated object, interaction part, and desired state transition for DT rollout generation. The LLM reasoner is queried through a structured JSON interface rather than inside the per-step control loop; planning and verification use deterministic decoding, while ontology enrichment uses a small nonzero temperature. To characterize the compute cost of the future-generation component, our mask-free video diffusion model was trained on a single NVIDIA A40 GPU with peak training memory of 44 GB; at inference, generating one 16-frame future clip with 20 denoising steps takes approximately 40–45 s in bf16 precision on the same hardware. These measurements indicate that the future-generation module operates in a practical single-GPU setting, although we do not claim a full efficiency benchmark. The LLM-based reasoner is outside the closed-loop policy execution path.

Observations and control. Policies operate from RGB observations and output continuous end-effector actions with

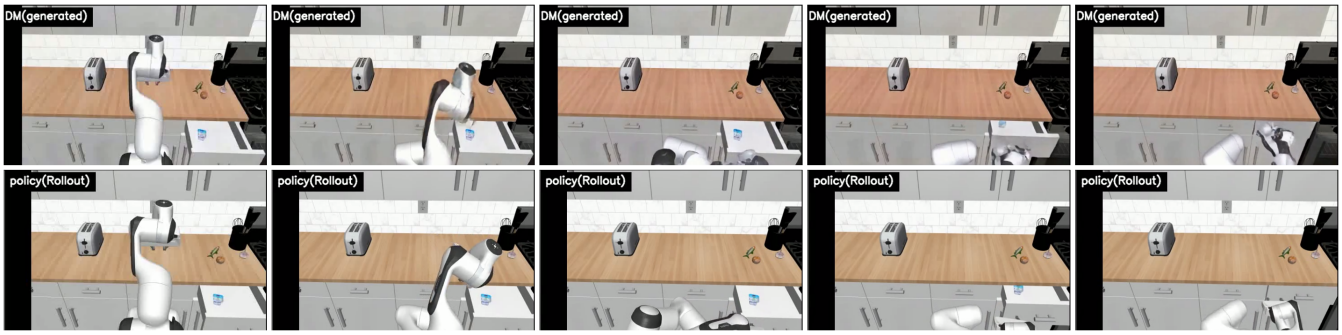

Fig. 3. Qualitative example on RoboCasa. Top: future sequence generated by the video diffusion model (VDM). Bottom: rollout of the robot policy conditioned on that generated future.

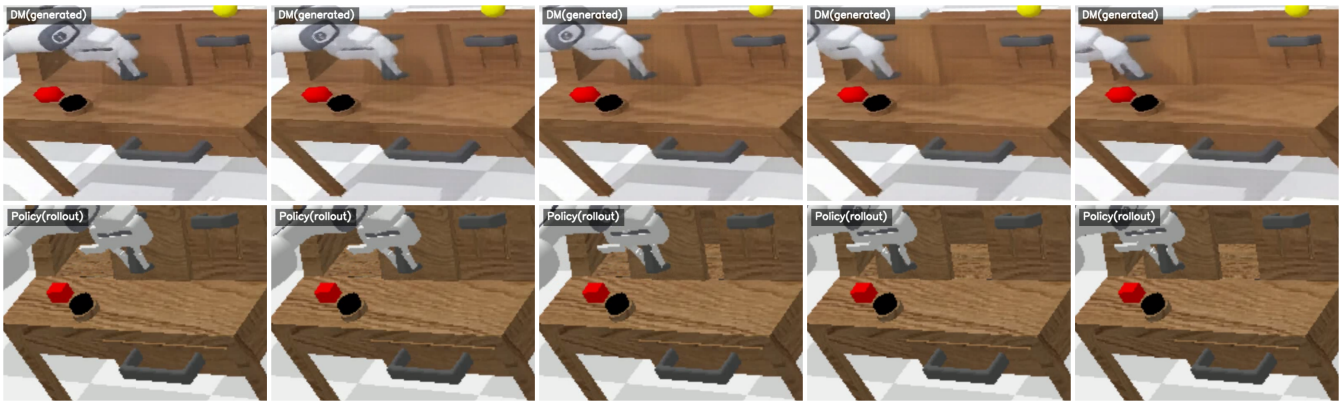

Fig. 4. Qualitative examples on CALVIN. Top: future sequence generated by the video diffusion model (VDM). Bottom: rollout of the robot policy conditioned on that generated future.

a gripper command. At each control step t, the controller receives the observation history $o_{t-K:t}$ and the future-conditioning vector g_t, and outputs an action $a_t \in \mathbb{R}^7$ consisting of a 6-DoF end-effector pose delta and a gripper command.

Future sources and training. We compare ground-truth futures extracted from demonstrations and generated futures produced by our framework. Generated futures are constructed in two stages: a robot-free DT rollout specifies the intended object motion, and a mask-free video diffusion inpainting model generates a robot-containing future clip consistent with that rollout. We train the VDM on paired DT/robot clips using fixed-length clips and first-frame anchoring. We then evaluate a BC-only controller and a BC+RL controller, with a future-conditioned SFP baseline for comparison. For each controller, we train three independent models with random seeds 42, 43, and 44.

Metrics. We report task success rate (%) under NoFuture, GTFuture, GenFuture, and WrongFuture. Each evaluation episode runs for at most 200 control steps.

B. Baselines and Ablations

We evaluate the following baselines and ablations:

- **Controllers:** BC-only, BC+RL, and a future-conditioned SFP comparison baseline.
- **NoFuture:** no future-conditioning signal.
- **GTFuture:** ground-truth future clips at test time.
- **GenFuture:** generated futures from our framework at test time.
- **WrongFuture:** intentionally mismatched futures to verify that gains are not due simply to extra input.

C. Results

Summary of quantitative results. Across both benchmarks, our main BC+RL instantiation is strongest overall, while BC-only remains competitive on some easier settings such as CALVIN `open_drawer`. The future-conditioned SFP baseline benefits from the same interface but remains below BC+RL. Generated future conditioning improves over NoFuture, whereas WrongFuture degrades sharply. This suggests that task-consistent future conditioning can support policy execution and reinforcement-learning-based adaptation, whereas mismatched future signals can be harmful.

Exploration efficiency during RL fine-tuning. To test whether generated future conditioning improves RL adaptation beyond final execution alone, we analyze BC+RL learning curves averaged across 8 CALVIN tasks over the full 140k RL environment steps. Figure 5 shows that GTFuture improves fastest and reaches the highest overall performance, GenFuture also improves earlier and to a higher level than NoFuture, and WrongFuture remains at zero throughout training. This provides broader evidence, beyond a single

(a) CALVIN: SFP

Task	NoF	GT	Gen	Wrong
open_drawer	19.0 ± 4.7	31.5 ± 1.2	29.1 ± 2.8	0.0 ± 0.0
turn_on_lightbulb	32.7 ± 4.9	59.8 ± 3.1	53.3 ± 1.8	0.0 ± 0.0
turn_on_led	9.7 ± 3.1	32.7 ± 3.0	22.6 ± 1.6	0.0 ± 0.0
push_into_drawer	8.9 ± 3.5	18.7 ± 2.9	13.9 ± 2.9	0.0 ± 0.0

(b) CALVIN: BC+RL

Task	NoF	GT	Gen	Wrong
open_drawer	87.1 ± 2.3	100.0 ± 0.0	98.0 ± 1.1	0.0 ± 0.0
close_drawer	85.8 ± 2.9	96.1 ± 1.9	89.2 ± 2.4	0.0 ± 0.0
turn_on_lightbulb	89.1 ± 2.1	100.0 ± 0.0	93.0 ± 1.1	0.0 ± 0.0
turn_off_lightbulb	91.1 ± 1.7	100.0 ± 0.0	94.0 ± 1.3	0.0 ± 0.0
turn_on_led	66.9 ± 2.1	100 ± 0.0	91.1 ± 2.2	0.0 ± 0.0
turn_off_led	64.1 ± 3.2	100 ± 0.0	92.3 ± 1.6	0.0 ± 0.0
push_into_drawer	31.8 ± 3.1	66.7 ± 1.9	50 ± 2.1	0.0 ± 0.0
move_slider_left	11.8 ± 1.1	50 ± 1.2	39 ± 3.7	0.0 ± 0.0

(c) CALVIN: BC-only

Task	NoF	GT	Gen	Wrong
open_drawer	73.3 ± 1.2	95.0 ± 2.2	85.7 ± 3.1	0.0 ± 0.0
close_drawer	74.1 ± 1.2	92.3 ± 1.4	90.1 ± 2.2	0.0 ± 0.0
turn_on_lightbulb	71.7 ± 1.9	90.0 ± 2.2	84.0 ± 1.7	0.0 ± 0.0
turn_off_lightbulb	74.2 ± 1.9	92.1 ± 1.5	87.1 ± 2.9	0.0 ± 0.0
turn_on_led	49.3 ± 2.3	62.2 ± 2.3	60.3 ± 1.9	0.0 ± 0.0
turn_off_led	47.9 ± 3.8	69.1 ± 1.2	61.2 ± 3.5	0.0 ± 0.0
push_into_drawer	8.7 ± 1.3	31.2 ± 2.8	18.1 ± 2.3	0.0 ± 0.0
move_slider_left	10.4 ± 3.1	35.8 ± 2.1	20.1 ± 2.8	0.0 ± 0.0

(d) RoboCasa: CloseDrawer

Method	NoF	GT	Gen	Wrong
SFP	36.7 ± 2.5	58.3 ± 2.1	49.7 ± 2.3	0.0 ± 0.0
BC-only	52.3 ± 2.1	74.7 ± 2.5	66.3 ± 2.1	0.0 ± 0.0
BC+RL	61.7 ± 2.5	82.3 ± 2.1	75.7 ± 2.3	0.0 ± 0.0

(e) RoboCasa: OpenSingleDoor

Method	NoF	GT	Gen	Wrong
SFP	0.0 ± 0.0	5.1 ± 1.2	0.0 ± 0.0	0.0 ± 0.0
BC-only	33.1 ± 2.8	63.2 ± 1.9	56.3 ± 3.3	0.0 ± 0.0
BC+RL	46.1 ± 1.3	73.1 ± 1.7	66.2 ± 1.2	0.0 ± 0.0

TABLE I

SUCCESS RATE (%) UNDER FOUR FUTURE CONDITIONS: NOFUTURE (NOF), GTFUTURE (GT), GENFUTURE (GEN), AND WRONGFUTURE (WRONG). RESULTS ARE AVERAGED OVER 3 TRAINING SEEDS.

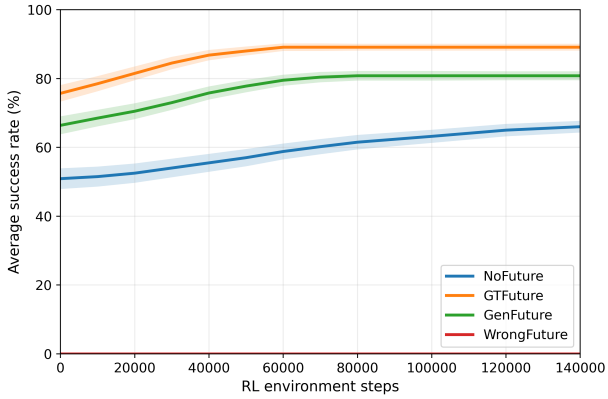

Fig. 5. Average BC+RL learning curves across 8 CALVIN tasks under four future conditions. Curves show mean success rate over RL environment steps, averaged across tasks, and shaded regions indicate standard deviation over 3 seeds. GTFuture improves fastest and reaches the highest average performance, GenFuture also improves earlier and to a higher level than NoFuture, while WrongFuture remains at zero throughout training. Final values are consistent with the average BC+RL results in Table I.

task, that informative future conditioning can improve RL fine-tuning efficiency, while mismatched future signals can strongly hinder adaptation.

Interpretation as structured exploration. Accurate futures accelerate adaptation most, realistic but imperfect futures remain useful, the absence of future guidance slows learning, and mismatched futures can fully misguide behavior.

Limitations and failure modes. Our study is limited to simulation and does not evaluate sim-to-real transfer or real-robot performance. The task-grounding stage uses privileged simulator state at task initialization rather than raw sensory reconstruction alone, and we do not provide a dedicated quantitative study of digital-twin or video-generation fidelity. Failures usually arise when the generated future contains inaccurate contact geometry or misleading interaction cues.

V. CONCLUSION

We presented a simulation study of future-conditioned control for multi-step robot manipulation. Our main controller instantiation is a behavior-cloning pipeline with reinforcement-learning fine-tuning, while a future-conditioned SFP model is included as a comparison baseline. Our framework combines LLM-guided task grounding, a robot-free digital-twin rollout, and mask-free diffusion-based robot inpainting to generate short-horizon future clips that are used as conditioning signals for control. Across RoboCasa and CALVIN, task-consistent future conditioning improves performance over no-future conditioning in our experiments, while mismatched futures are harmful, and our BC+RL instantiation achieves the strongest overall results. In addition, an average BC+RL learning-curve analysis across 8 CALVIN tasks shows that GTFuture improves fastest, GenFuture also improves earlier and to a higher level than NoFuture, and WrongFuture remains at zero throughout training.

These results support the view that short-horizon future videos can serve as useful structured priors for policy execution and RL adaptation under imperfect future predictions.

ACKNOWLEDGMENT

The authors thank their collaborators and reviewers for helpful feedback.

REFERENCES

[1] S. Jiang, X. Fang, N. Roy, T. Lozano-Perez, L. P. Kaelbling, and S. Ancha, "Streaming flow policy: Simplifying diffusion/flow-matching policies by treating action trajectories as flow trajectories," *arXiv preprint arXiv:2505.21851*, 2025.

[2] D. Han *et al.*, "A survey on deep reinforcement learning algorithms for robotic manipulation," *Sensors*, 2023.

[3] Í. Elguea-Aguinaco *et al.*, "A review on reinforcement learning for contact-rich robotic manipulation tasks," *Robotics and Computer-Integrated Manufacturing*, 2023.

[4] C. Finn and S. Levine, "Deep visual foresight for planning robot motion," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[5] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," *arXiv preprint arXiv:1811.04551*, 2019.

[6] D. Ha and J. Schmidhuber, "World models," *arXiv preprint arXiv:1803.10122*, 2018.

[7] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," *arXiv preprint arXiv:1912.01603*, 2019.

[8] A. Brohan, N. Brown, J. Carbajal *et al.*, "Rt-1: Robotics transformer for real-world control at scale," in *Robotics: Science and Systems (RSS)*, 2023.

[9] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K. Lee, S. Levine, Y. Lu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, M. Yan, A. Zeng *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[10] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black *et al.*, "Octo: An open-source generalist robot policy," in *Robotics: Science and Systems (RSS)*, 2024.

[11] M. D. Tezerjani, M. Khoshnazar, M. Tangestanizadeh, A. Kiani, and Q. Yang, "A survey on reinforcement learning applications in slam," *arXiv preprint arXiv:2408.14518*, 2024.

[12] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[13] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *International Conference on Machine Learning (ICML)*, 2022.

[14] Y. Luo and Y. Du, "Grounding video models to actions through goal-conditioned exploration," in *International Conference on Learning Representations (ICLR)*, 2025.

[15] Q. Bu, J. Zeng, L. Chen, Y. Yang, G. Zhou, J. Yan, P. Luo, H. Cui, Y. Ma, and H. Li, "Closed-loop visuomotor control with generative expectation for robotic manipulation," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[16] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[17] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, "Video diffusion models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[18] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "Repaint: Inpainting using denoising diffusion probabilistic models," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[19] Z. Yang, J. Teng, W. Zheng, M. Ding, S. Huang, J. Xu, Y. Yang, W. Hong, X. Zhang, G. Feng, D. Yin, X. Gu, Y. Zhang, W. Wang, Y. Cheng, T. Liu, B. Xu, Y. Dong, and J. Tang, "Cogvideox: Text-to-video diffusion models with an expert transformer," *arXiv preprint arXiv:2408.06072*, 2024.

[20] Y. Bian, Z. Zhang, X. Ju, M. Cao, L. Xie, Y. Shan, and Q. Xu, "Videopainter: Any-length video inpainting and editing with plug-and-play context control," *arXiv preprint arXiv:2503.05639*, 2025.

[21] I. Kapelyukh, Y. Ren, I. Alzugaray, and E. Johns, "Dream2Real: Zero-shot 3D object rearrangement with vision-language models," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 4796–4803.

[22] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, C. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, B. Ichter, J. Ibarz, A. Irpan, E. Jang, R. J. Ruano, D. Kalashnikov, S. Levine *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[23] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Robotics: Science and Systems (RSS)*, 2023.

[24] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," *arXiv preprint arXiv:2109.12098*, 2021.

[25] ——, "Peract: Multi-task 6d robotic manipulation learning from perceiver-actor," *arXiv preprint arXiv:2209.05451*, 2022.

[26] Y. Jiang, A. Gupta, A. Zeng, P. Florence, M. Ahn, Y. Lu, J. Ibarz, B. Ichter, A. Wahid, K. Kumar, J. Xu, S. Levine, and J. Tompson, "Vima: General robot manipulation with multimodal prompts," *arXiv preprint arXiv:2210.03094*, 2022.

[27] N. M. M. Shafiullah, C. Paxton, and L. Pinto, "Behavior transformers: Cloning k modes with one stone," *arXiv preprint arXiv:2206.11251*, 2022.

[28] S. Ross, G. Gordon, and J. A. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[29] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.