Fetch My Beer: Synthetic-to-Real Hierarchical Policy for Smooth Pick-and-Place

Yingyue Li, Chenyangguang Zhang, Ruida Zhang, Bowen Fu, Guangyao Zhai, and Xiangyang Ji

***Abstract*—Many real-world robotic applications require dynamically sensitive manipulation, where success depends not only on reaching a target state but on maintaining stable object dynamics throughout execution. We study the stable transport of liquid-filled containers, where a robot must move objects to target locations while suppressing sloshing and preventing spillage. Unlike conventional pick-and-place, this task imposes stringent requirements on motion smoothness and trajectory-level stability, exposing clear limitations in existing systems. Specifically, fluid simulation remains too costly for online reinforcement learning; human teleoperation introduces unintended accelerations that induce sloshing during imitation learning; and current policy pipelines optimize for task completion rather than dynamic stability. We propose a synthetic-to-real framework coupling physically validated data generation with a hierarchical, diffusion-based controller. The scalable data pipeline synthesizes grasps, filters unstable poses via a vision-language model, and validates transport trajectories through fluid simulation. The policy is organized with a high-level module that translates language and visual observations into SE(3) control targets, and a latent diffusion controller that first plans efficiently in a compact latent space and then decodes dense action chunks, enabling the high control frequency needed for smooth and stable motion. Extensive experiments show our system outperforms state-of-the-art manipulation policies in transport smoothness and dynamic stability.**

***Index Terms*—Deep learning in grasping and manipulation, learning from demonstration, machine learning for robot control.**

I. Introduction

RECENT advances in robot manipulation [1], [2], [3], [4], [5], [6], [7] have enabled remarkable progress in tabletop manipulation such as pick-and-place, rearrangement, and tool use. However, many real-world tasks require dynamically sensitive manipulation, where success depends not only on reaching a goal pose, but also on maintaining stable object dynamics throughout execution. Representative examples include transporting liquid-filled containers, carrying fragile or deformable objects, and manipulating items with unstable contents, which is highly relevant to service robotics, laboratory automation, and industrial delivery scenarios. In these tasks, even small undesired accelerations, jitter, or abrupt corrective motions can induce irreversible dynamic disturbances, leading to failure despite correct final positioning. In this work, we study stable robotic transport of liquid-filled containers, where a robot must move objects to target locations while suppressing sloshing and preventing spillage. Compared with conventional pick-and-place tasks, such dynamically sensitive manipulation imposes substantially stricter constraints on motion smoothness and stability, requiring joint reasoning over semantics, geometry, and fine-grained control (See Fig. 1).

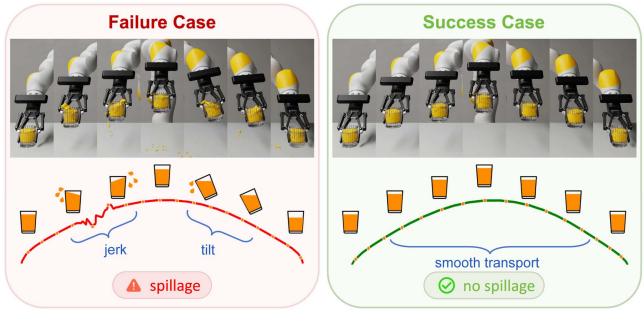


Fig. 1. In dynamically sensitive tasks such as liquid transport, abrupt acceleration and excessive tilt lead to sloshing and spillage, whereas smooth, temporally consistent motion enables stable, spill-free transport.

Despite its practical importance, stable and smooth robotic manipulation remains highly challenging for existing robot learning systems. For Reinforcement Learning (RL) methods, the computational expense of fluid simulation poses a severe bottleneck. Because computing fluid dynamics is significantly slower than rigid-body simulation, performing the millions of online, interactive steps required for RL convergence is computationally prohibitive. Conversely, Imitation Learning (IL) allows for offline training but faces a fundamental difficulty in data collection. Human teleoperation for dynamically sensitive tasks is highly error-prone, since even small unintended accelerations can induce severe sloshing or instability. Although simulation-based data generation provides a scalable alternative, existing pipelines [8], [9], [10], [11], [12], [13] primarily optimize for rigid-body task completion and geometric feasibility, while overlooking motion stability and smoothness. On the

Received 26 June 2026; accepted 19 August 2026. Date of publication 10 September 2026; date of current version 2 October 2026. This article was recommended for publication by Associate Editor Y. Xiang and Editor M. Vincze upon evaluation of the reviewers' comments. This work was supported by the National Science and Technology Major Project under Grant 2025ZD1606303. *(Yingyue Li, Chenyangguang Zhang, and Ruida Zhang contributed equally to this work.) (Corresponding author: Xiangyang Ji.)*

Yingyue Li, Ruida Zhang, and Xiangyang Ji are with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: yingyue-21@mails.tsinghua.edu.cn; zhangrd23@mails.tsinghua.edu.cn; xyji@tsinghua.edu.cn).

Chenyangguang Zhang is with ETH Zurich, 8092 Zurich, Switzerland (e-mail: chenyangguang.zhang@inf.ethz.ch).

Bowen Fu is with Beijing Xiaomi Robot Technology Company, Ltd, Beijing 100176, China (e-mail: fubowen@xiaomi.com).

Guangyao Zhai is with the Technical University of Munich, 80333 Munich, Germany (e-mail: guangyao.zhai@tum.de).

Our project page: https://fetch-my-beer.github.io/.

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2026.3732928, provided by the authors.

Digital Object Identifier 10.1109/LRA.2026.3732928

2377-3766 © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Tsinghua University. Downloaded on October 19,2026 at 12:54:23 UTC from IEEE Xplore. Restrictions apply.

policy side, recent diffusion-based visuomotor policies [3], [14], [15], [16], [17] and Vision-Language-Action (VLA) models [4], [5], [6], [7] perform well on general manipulation benchmarks, but primarily optimize geometric accuracy rather than dynamic stability. In particular, dynamically sensitive manipulation tasks require policies to jointly perform semantic reasoning, long-horizon motion planning, and high-frequency smooth control under stringent stability constraints. This multi-scale requirement remains difficult for existing flat, end-to-end architectures to balance. As a result, enabling reliable and stable manipulation requires not only physically grounded data generation pipelines, but also control architectures explicitly designed for temporally coherent and stability-aware manipulation.

To address these challenges, we propose a comprehensive synthetic-to-real framework that combines physically validated synthetic data generation with a hierarchical, diffusion-based control architecture for stable and smooth robotic manipulation. On the data generation side, we develop a scalable pipeline in NVIDIA Isaac Sim and Isaac Lab [18] to automatically produce high-quality demonstrations across diverse scenes and container categories. Specifically, we generate diverse grasp candidates using state-of-the-art grasp models and leverage a Vision-Language Model (VLM) to filter out physically infeasible or geometrically unstable poses. We then compute smooth trajectories using motion planning and rigorously validate them through fluid simulation. This ensures that the collected dataset satisfies both geometric feasibility and stringent trajectory-level stability constraints. On the policy side, we introduce a hierarchical control framework that decouples semantic task reasoning from high-frequency motion generation. At the high level, a keyframe policy infers task-relevant SE(3) sub-goals from language instructions and multi-view RGB-D observations. Conditioned on these sub-goals, the low-level controller employs a latent diffusion architecture: compact latent action sequences are efficiently generated in a lower-dimensional space, then subsequently refined into dense, executable trajectories via a decoder. This hierarchical design promotes temporally coherent, smooth motion generation while maintaining high-frequency responsiveness.

To evaluate manipulation under fluid-induced dynamics, we introduce a simulation benchmark and metrics that prioritize trajectory-level stability over mere task success. Our benchmark assesses policy generalization across diverse object geometries, workspace layouts, and visual shifts. To quantify performance, we introduce metrics that jointly measure task success, trajectory smoothness, container orientation stability, and worst-case tilt. Experiments demonstrate that our method consistently outperforms strong hierarchical diffusion and VLA baselines across both in-domain and out-of-domain settings. Furthermore, trained solely on synthetic demonstrations, our policy achieves effective zero-shot sim-to-real transfer without any real-world fine-tuning.

Our main contributions are summarized as follows:

- We develop a scalable synthetic data generation pipeline with physics-based fluid validation, enabling efficient collection of dynamically stable manipulation demonstrations and reliable sim-to-real transfer.
- We introduce a simulation benchmark together with evaluation metrics that capture both task success and trajectory-level dynamic stability, enabling systematic evaluation under in-domain and out-of-domain settings.
- We design a hierarchical latent diffusion framework that decouples semantic reasoning from high-frequency motion generation, achieving stable and smooth manipulation in simulation and zero-shot real-world scenarios.

II. Related Work

A. Dynamically Sensitive Manipulation

Manipulating dynamically sensitive objects is a long-standing problem in robotics, where task success depends on both geometric accuracy and stable physical evolution. Early studies on liquid transport [19], [20], [21], [22], [23] primarily relied on model-based control, trajectory optimization, and vibration suppression techniques to reduce fluid oscillation. While these methods demonstrate strong performance in carefully modeled settings, they often require accurate fluid dynamics priors, handcrafted controllers, or fixed robot configurations, limiting their generalization to unstructured settings. This has motivated increasing interest in learning-based approaches for stability-aware manipulation.

B. Visuomotor Policy and Hierarchical Robot Control

Behavior cloning (BC) methods [1], [2], [3] learn policies from expert demonstrations, enabling robots to directly map visual observations and language instructions to continuous actions. Large VLA models [4], [5], [6], [7] further improve task generalization through vision-language pretraining. Direct dense action prediction, however, can be computationally expensive and temporally inconsistent, which is problematic when small action errors disturb object dynamics. Hierarchical control mitigates this issue by decomposing manipulation across temporal abstraction levels. Prior works have explored hierarchical learning through sub-tasks, sub-goals, and multi-stage policies [24], [25], [26], [27], which improve long-horizon reasoning and simplify low-level action generation. We follow this principle and use event-labeled keyframes for task-level reasoning and latent diffusion for smooth, high-frequency control.

C. Synthetic Data Generation

Large-scale demonstration data is essential for training robust visuomotor policies, yet real-world collection of dynamically stable manipulation trajectories is expensive and difficult to scale. Recent robotic simulation frameworks and synthetic data generation benchmarks, such as RoboTwin [9], [10] and ManiSkill [11], [12], [13], have enabled scalable demonstration generation through scripted expert policies, dense-reward RL, or human teleoperation. However, most pipelines emphasize rigid-body completion and collision-free motion rather than dynamic stability along the full trajectory. Our pipeline additionally applies physics-based validation, retaining trajectories that are both geometrically feasible and dynamically stable.

III. Simulation

Training stable and precise manipulation policies requires a large volume of high-quality demonstrations that are kinematically feasible, smooth, and dynamically safe. Real-world teleoperation is expensive and unreliable for this purpose, as

Authorized licensed use limited to: Tsinghua University. Downloaded on October 19,2026 at 12:54:23 UTC from IEEE Xplore. Restrictions apply.

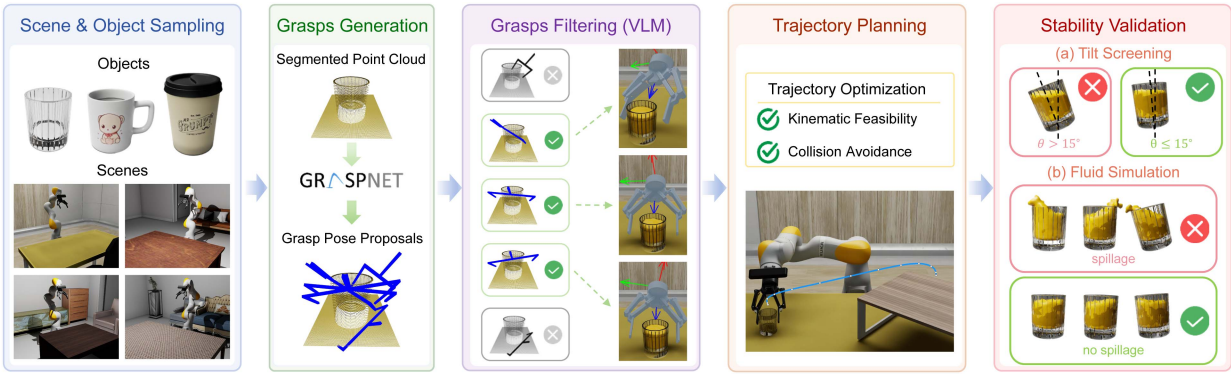


Fig. 2. Stability-aware synthetic data generation pipeline. We sample diverse objects and scenes, generate 6-DoF grasp candidates from segmented point clouds, and use a vision-language model to score each rendered grasp proposal with a short feasibility reason. Selected grasps are passed to a smooth trajectory planner, which enforces kinematic feasibility and collision avoidance. Candidate trajectories are then filtered by tilt screening and physics-based fluid simulation to remove unstable or spill-prone motions.

small control errors can easily destabilize the manipulated object. While simulation offers a scalable alternative, existing pipelines primarily target rigid-body collision avoidance and goal reaching. For dynamically sensitive tasks, such geometric checks are insufficient: trajectories that reach the goal can still fail due to abrupt acceleration or invalid intermediate poses. We therefore build a stability-aware synthetic data generation pipeline in NVIDIA Isaac Sim [18], shown in Fig. 2. For each scene configuration, the pipeline proposes grasp candidates, filters implausible ones via a VLM, plans and optimizes grasp-to-place trajectories with cuRobo [28], and validates them through rule-based tilt screening and physics-based fluid simulation. This pipeline enables scalable collection of high-quality demonstrations that satisfy semantic consistency, kinematic feasibility, and trajectory-level dynamic stability.

A. *Grasping Proposals*

We use AnyGrasp [29], a state-of-the-art grasp pose estimation model, to generate diverse 6-DoF grasp hypotheses from rendered scene point clouds. Given the target object geometry, the grasp model produces a ranked set of candidate gripper poses that covers multiple approach directions and contact regions. Retaining multiple candidates is important for stable manipulation: a grasp that is reachable and force-closure feasible may still lead to poor downstream motion if it constrains the wrist, induces unfavorable object poses during transport, or leaves little margin for smooth placement. We therefore keep a diverse candidate set rather than committing to the top geometric grasp directly. This design allows the following stages to select grasps according to task-level plausibility and trajectory feasibility, not only local contact geometry.

B. *VLM-Based Filtering*

Purely geometric grasp models do not account for whether a grasp is suitable for stable downstream motion, and their confidence scores may not reflect transport feasibility in a specific scene. We therefore use Qwen3-VL [30] as a feasibility filter before motion planning. Since the VLM operates on 2D inputs, we render each 6-DoF grasp by projecting a simplified gripper mesh onto the scene image, thereby explicitly visualizing the proposed gripper-object interaction. Given this composite image, the VLM predicts a confidence score with a brief justification. This stage filters out candidates that are visually implausible or unsuitable for the task, such as grasps that collide with the table, insert the gripper into the container, approach the object from an infeasible direction, contact unstable regions, or result in an unfavorable object pose for subsequent transport. The remaining candidates are sorted by VLM confidence and passed to the motion planning stage. In this way, the pipeline combines the geometric coverage of grasp proposal networks with the visual reasoning and commonsense feasibility assessment of a VLM before costly trajectory execution and stability validation.

C. *Smooth Path Planning With Stability-Aware Filtering*

To obtain high-quality robot end-effector trajectories that satisfy the stringent smoothness requirements for manipulating liquid-filled containers, we employ cuRobo [28], a CUDA-accelerated library, as an efficient motion generation backend for inverse kinematics and trajectory optimization. Given the grasp candidates filtered and ranked by VLM, we randomly sample multiple candidate placement poses on the target platform for each selected grasp and use cuRobo to solve the corresponding grasp-to-place motions. Among all these feasible motions, we select the trajectory with the best optimization quality, considering IK feasibility, collision avoidance, and trajectory cost.

To ensure that the collected data satisfy the strict smoothness and stability requirements of stable and smooth manipulation, we further apply two complementary filtering mechanisms: a rule-based screening criterion and physics-based fluid simulation. The rule-based screening serves as a lightweight preliminary filter to quickly eliminate clearly unsuitable trajectories during execution. Specifically, we monitor a dedicated Z-axis cup angle metric and reject trajectories in which the cup deviates from the vertical direction by more than an experimentally determined threshold of 15 degrees. This criterion provides an efficient way to remove motions that are highly likely to cause spillage during transport. Beyond the tilt-based geometric heuristic, we further evaluate the remaining trajectories with particle-based fluid simulation in Isaac Sim. This step is designed to capture spill-prone motions that cannot be detected by cup orientation alone. In particular, a trajectory may keep the container nearly upright while still inducing strong liquid sloshing due to abrupt velocity changes, high accelerations, or jerky

Authorized licensed use limited to: Tsinghua University. Downloaded on October 19,2026 at 12:54:23 UTC from IEEE Xplore. Restrictions apply.

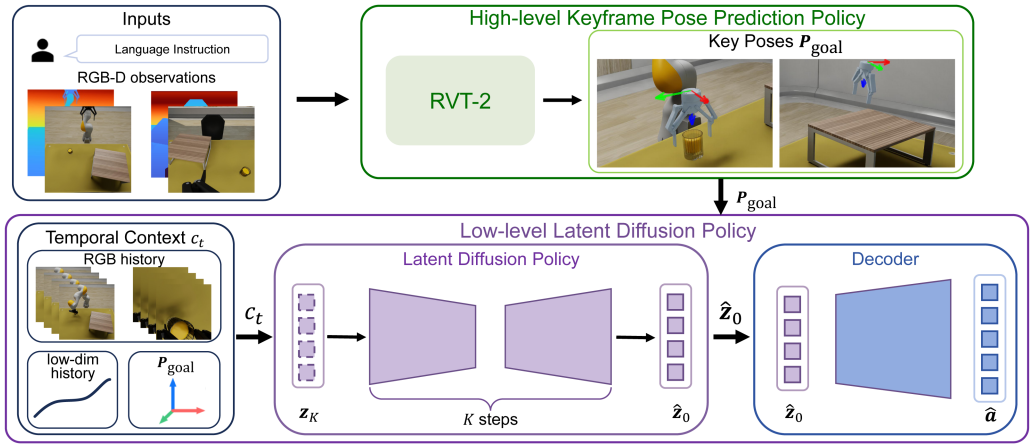


Fig. 3. Architecture of the proposed hierarchical latent diffusion policy. The high-level policy predicts task-relevant SE(3) sub-goals $\mathbf{P}_{\text{goal}}$ from language instructions and multi-view RGB-D observations. Conditioned on the predicted target pose, recent visual observations, and proprioceptive state history, the low-level latent diffusion policy iteratively denoises in the learned latent action space to generate a latent action sequence $\hat{\mathbf{z}}_0$. A lightweight decoder then maps $\hat{\mathbf{z}}_0$ into dense executable action chunks $\hat{\mathbf{a}}$ for smooth and high-frequency execution.

motions during transport. Such dynamic effects are difficult to identify using rule-based tilt screening, but they are critical for stable liquid manipulation. We therefore use the Position Based Dynamics (PBD) [31] solver in Isaac Sim as a physics-based validation stage to reject trajectories that complete the geometric task yet cause excessive liquid oscillation or spillage. Only trajectories that pass both filters are stored as demonstrations. As a result, the dataset contains motions that are geometrically feasible, smooth at the trajectory level, and dynamically stable under the fluid-sensitive setting.

IV. Hierarchical Latent Diffusion Policy

We propose a hierarchical control framework for stable and smooth pick-and-place tasks, as illustrated in Fig. 3. The key motivation is that dynamically sensitive manipulation involves heterogeneous requirements across temporal scales. Task-level reasoning determines where to grasp and place from language and multi-view observations, while execution-level control must generate high-frequency, temporally coherent actions without abrupt corrections. Coupling these requirements into a monolithic observation-to-action policy makes it difficult to balance low-frequency semantic reasoning with high-frequency smooth control. Moreover, directly applying diffusion to dense action chunks in the original action space can be computationally expensive and may introduce temporally inconsistent low-level motions, which are undesirable for stability-sensitive execution. We therefore decompose control into a high-level keyframe policy and a low-level latent diffusion policy. The high-level policy predicts SE(3) sub-goals for the current manipulation stage. Conditioned on each sub-goal, the low-level policy first generates compact latent action sequences through diffusion and then decodes them into dense executable action chunks for stable execution.

A. High-Level Keyframe Pose Prediction Policy

The high-level policy π_H maps language instructions l and multi-view RGB-D observations $\mathcal{O}_t$ to the next keyframe end-effector pose $\mathbf{P}_{\text{goal}} \in \text{SE}(3)$, which represents task-level sub-goals such as grasping or placing poses:

$$\mathbf{P}_{\text{goal}} = \pi_H(\mathcal{O}_t, l). \tag{1}$$

During data generation, we directly record the end-effector poses at the successful grasp and placement events as the two keyframe annotations, requiring neither manual labeling nor velocity-based post-processing. We employ RVT-2 [1] as the high-level policy and retain its original architecture. RVT-2's design features a virtual-view re-rendering mechanism that enhances camera-viewpoint generalization, alongside a multi-stage refinement pipeline that improves pose predictions. The predicted sub-goals are passed to the low-level policy, decoupling semantic reasoning from smooth motion generation.

B. Low-Level Latent Diffusion Policy

The primary objective of the low-level policy is to translate the sparse SE(3) sub-goals predicted by the high-level policy into smooth, high-frequency end-effector trajectories. A straightforward solution is to employ the standard Diffusion Policy [3] to directly generate dense action chunks in the raw action space. However, its iterative denoising process is computationally costly for high-frequency control, and small inconsistencies across successive predicted chunks can lead to oscillatory motions, increasing the risk of fluid spillage.

Inspired by recent latent-space diffusion [15], [25], we propose a Latent Diffusion Policy that separates efficient motion generation from dense trajectory reconstruction. Specifically, we train an action autoencoder that compresses a dense action chunk into a lower-dimensional latent action sequence along both the temporal and action-feature dimensions, and reconstructs it back into a dense, executable trajectory. The diffusion policy generates a motion plan in this compact latent space, reducing inference latency compared to raw-action diffusion. Then, a lightweight decoder decodes the predicted latent representations into dense, high-frequency action chunks for execution. This design preserves the generative flexibility of diffusion models

Authorized licensed use limited to: Tsinghua University. Downloaded on October 19,2026 at 12:54:23 UTC from IEEE Xplore. Restrictions apply.

while improving both computational efficiency and temporal consistency for stable manipulation.

Action Autoencoder: The action autoencoder is formulated as a variational autoencoder (VAE) that compresses dense action chunks along both the temporal and action-feature dimensions, constructing a compact latent space for downstream diffusion-based trajectory generation. Given a dense action chunk $\mathbf{a} \in \mathbb{R}^{T\times D_a}$, the encoder $\mathcal{E}_\psi$ predicts the parameters of a diagonal Gaussian posterior:

$$q_\psi(\mathbf{z} \mid \mathbf{a}) = \mathcal{N}\left(\boldsymbol{\mu}_\psi(\mathbf{a}), \mathrm{diag}(\boldsymbol{\sigma}^2_\psi(\mathbf{a}))\right), \tag{2}$$

where $\mathbf{z} \in \mathbb{R}^{T'\times D_z}$, $T' < T$, and $D_z < D_a$. During training, a latent action sequence $\tilde{\mathbf{z}}$ is sampled from this posterior using the reparameterization trick and then reconstructed by a lightweight decoder:

$$\hat{\mathbf{a}} = \mathcal{D}_\phi(\tilde{\mathbf{z}}), \quad \hat{\mathbf{a}} \in \mathbb{R}^{T\times D_a}, \tag{3}$$

where $\mathcal{D}_\phi$ is implemented as a multilayer perceptron (MLP). The encoder and decoder are trained jointly with reconstruction and KL regularization losses:

$$\begin{aligned}\mathcal{L}_{\mathrm{AE}} = \ & \lambda_1 \left\|\mathbf{a} - \hat{\mathbf{a}}\right\|_1 + \lambda_2 \left\|\mathbf{a} - \hat{\mathbf{a}}\right\|_2^2 \\ & + \lambda_{\mathrm{KL}} D_{\mathrm{KL}}\left(q_\psi(\mathbf{z} \mid \mathbf{a}) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I})\right).\end{aligned} \tag{4}$$

The reconstruction terms encourage accurate recovery of dense executable trajectories, while the KL term regularizes the latent posterior toward a standard Gaussian prior, yielding a continuous latent space for diffusion-based generation.

Latent Diffusion Policy: We formulate action generation as a goal-conditioned diffusion process in a continuous latent action space. After training the action autoencoder, each demonstration action chunk $\mathbf{a}$ is encoded into a deterministic latent action representation as the clean target for diffusion by taking the posterior mean of the encoder:

$$\mathbf{z}_0 = \boldsymbol{\mu}_\psi(\mathbf{a}). \tag{5}$$

To guide the generation process of $\mathbf{z}_0$, the policy is conditioned on a multi-modal temporal context $\mathbf{c}$ that integrates visual perception, proprioceptive state history, and goal specification. Formally, the conditioning signal is defined as $\mathbf{c} = \{\mathbf{o}^{\mathrm{rgb}}_{t-h:t}, \mathbf{o}^{\mathrm{prop}}_{t-l:t}, \mathbf{P}_{\mathrm{goal}}\}$, where: (i) $\mathbf{o}^{\mathrm{rgb}}_{t-h:t}$ denotes a short-horizon sequence of RGB observations that provides recent visual feedback from the scene; (ii) $\mathbf{o}^{\mathrm{prop}}_{t-l:t}$ represents a longer history of robot proprioceptive states, capturing past motion dynamics and trajectory evolution; and (iii) $\mathbf{P}_{\mathrm{goal}} \in \mathrm{SE}(3)$ specifies the target end-effector pose predicted by the high-level policy. Together, these conditioning inputs enable the policy to adapt to short-term visual changes, maintain temporal consistency with the motion history, and generate actions directed toward the target pose.

We model the distribution of latent action sequences with a conditional Denoising Diffusion Probabilistic Model (DDPM) [32]. During training, we perturb the clean latent action sequence $\mathbf{z}_0$ defined in (5) with Gaussian noise at diffusion step k:

$$\mathbf{z}_k = \sqrt{\bar{\alpha}_k}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_k}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{6}$$

where $\bar{\alpha}_k$ denotes the cumulative noise schedule. A conditional noise prediction network $\boldsymbol{\epsilon}_\theta$, implemented as a 1D U-Net, takes the noisy latent sequence $\mathbf{z}_k$, the conditioning context $\mathbf{c}$, and the diffusion step k as input, and predicts the injected noise. The latent diffusion policy is trained with the standard noise prediction objective:

$$\mathcal{L}_{\mathrm{LDP}} = \mathbb{E}_{\mathbf{z}_0,\mathbf{c},\boldsymbol{\epsilon},k}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{z}_k, \mathbf{c}, k)\right\|_2^2\right]. \tag{7}$$

During inference, we sample an initial latent noise sequence $\mathbf{z}_K \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and iteratively denoise it conditioned on $\mathbf{c}$ for K steps to generate the predicted latent action sequence $\hat{\mathbf{z}}_0$. The resulting latent sequence is then fed into the decoder $\mathcal{D}_\phi$ to generate dense executable action chunks. Following ACT [2], we apply temporal ensemble across consecutive decoded chunks, where overlapping predictions for the same execution timestep are aggregated with exponentially decayed weights. This reduces chunk-boundary discontinuities and improves execution smoothness.

V. Experiments

A. Experimental Setup

Benchmark: We evaluate stable manipulation on a liquid transport benchmark, where a robot is required to grasp a liquid-filled container, transport it to a designated target platform, and place it successfully while suppressing sloshing and preventing spillage. We construct the benchmark using the synthetic data generation pipeline described in Section III. We use diverse cup assets from Isaac Sim and Objaverse [33], covering different materials, shapes, and visual appearances, together with randomized tabletop textures and scene settings. We introduce a raised target platform to increase spatial difficulty and evaluate stable transport across height variations. In total, we generate 12 scene variations with 100 demonstrations each, resulting in 1200 training episodes.

To evaluate both in-distribution performance and out-of-domain (OOD) robustness, we test all methods on 50 configurations for each of four distinct splits:

1) **In-domain**, following the training distribution;
2) **OOD-Object**, utilizing unseen cup instances;
3) **OOD-Scene**, using unseen tabletop textures and backgrounds; and
4) **OOD-Object+Scene**, combining both shifts to assess compound generalization.

For a fair comparison, all evaluations use the same fixed initializations and are reported over three random seeds.

Implementation Details: The high-level RVT-2 policy is trained on the generated keyframe annotations using the original training recipe. The low-level policy is trained sequentially in two stages.

Stage 1: Action Autoencoder: A CNN encoder compresses a 32×10 action chunk into an 8×8 latent action representation, which an MLP decoder reconstructs into raw actions (3D position, 6D rotation, and 1D gripper width). We optimize this model using the autoencoder loss $\mathcal{L}_{\mathrm{AE}}$ defined in (4) with AdamW for 600 epochs with a batch size of 64, a learning rate of 1×10^{-3}, and a weight decay of 1×10^{-4}.

Stage 2: Latent Diffusion Policy: The policy predicts these 8×8 latent action representations conditioned on four visual steps, a 20-step proprioceptive history, and the target keyframe pose. It uses a ResNet-18 image encoder and a 100-step DDIM scheduler. Optimization is performed using the latent diffusion loss $\mathcal{L}_{\mathrm{LDP}}$ defined in (7) with AdamW for 400 epochs with a batch size of 64, a learning rate of 1×10^{-4}, a weight decay of 1×10^{-6}, EMA, and a cosine decay schedule.

Authorized licensed use limited to: Tsinghua University. Downloaded on October 19,2026 at 12:54:23 UTC from IEEE Xplore. Restrictions apply.

Inference efficiency: On an RTX 3090, LDP uses 300 M parameters, 93.94 ms per inference, and 1.19 GB peak GPU memory, versus 330 M, 107.06 ms, and 1.33 GB for raw-action DP, reducing latency by 12.3% and memory usage by 10.5%.

Baselines: We compare our method against three representative baselines. First, **RVT-2 + DP** is a hierarchical baseline that shares our high-level module but replaces the low-level latent diffusion with a standard Diffusion Policy [3] operating in the raw action space. This baseline isolates the impact of our latent-space action generation design while keeping the hierarchical structure identical. Next, $\pi_{0.5}$ **(few-shot)** evaluates the pre-trained generalist VLA model $\pi_{0.5}$ [7] fine-tuned on 10% of our training data, assessing whether general VLA priors can rapidly adapt to dynamically sensitive manipulation. Finally, $\pi_{0.5}$ **(full data)** fine-tunes the same VLA model on the complete training dataset. This setting evaluates whether a monolithic VLA model can achieve comparable stability and smoothness when trained with the same task-specific demonstrations.

Metrics: To evaluate stable and smooth manipulation, we use four metrics that jointly measure task completion, object stability, and motion smoothness. To ensure meaningful and fair comparisons, the latter three metrics (OE, MTA, and TS) are computed exclusively over successful trials.

Success Rate: (SR, %) measures the percentage of trials in which the robot successfully grasps the cup, transports it to the target platform, and maintains the fluid loss below 5% of the total fluid particles.

Orientation Error: (OE, rad · s) evaluates the cumulative cup deviation from the upright direction during execution:

$$E_{\text{orient}} = \int_0^T \theta_{\text{tilt}}(t)\,\mathrm{d}t, \tag{8}$$

where $\theta_{\text{tilt}}(t)$ is the angle between the cup's central axis and the global vertical axis at time t, and T is the task duration. Lower OE indicates more consistent maintenance of the upright pose throughout transport.

Maximum Tilt Angle: (MTA, °) measures the worst-case cup deviation during execution:

$$\theta_{\max} = \max_{t\in[0,T]} \theta_{\text{tilt}}(t). \tag{9}$$

This metric captures transient instability that may be averaged out or missed by the cumulative orientation error.

Trajectory Smoothness: (TS, $\mathrm{m}^2/\mathrm{s}^6$) measures the average squared jerk of the end-effector trajectory:

$$J = \frac{1}{T}\int_0^T \|\dddot{\mathbf{x}}(t)\|_2^2\,\mathrm{d}t, \tag{10}$$

where $\dddot{\mathbf{x}}(t)$ denotes the end-effector jerk. Lower TS indicates smoother motion with fewer abrupt acceleration changes, which is critical for reducing liquid sloshing.

B. Simulation Evaluation

We first evaluate the proposed framework against the baselines in simulation, assessing both in-domain performance and out-of-domain robustness across four evaluation splits.

In-domain Evaluation: As shown in Table I, our method achieves the best overall performance among all methods. Compared with RVT-2 + DP, which shares the same high-level module but generates actions in the raw action space, our method improves the success rate by 10.49%. It also achieves lower OE, MTA, and TS, indicating better upright-pose maintenance and reduced transient instability during transport. These results show that generating low-level motions in a compact latent space is more effective than directly diffusing dense raw actions for stable manipulation. The VLA baselines perform substantially worse in this dynamically sensitive setting. Although full-data fine-tuning improves $\pi_{0.5}$ over the few-shot variant, its success rate remains far below that of the hierarchical methods. This indicates that general VLA priors alone are insufficient for stable manipulation, and that directly learning end-to-end actions remains challenging even with task-specific demonstrations. Qualitative rollouts of our policy, provided in the attached video and Fig. 4(a), demonstrate its ability to transport the container with minimal sloshing while maintaining a consistent upright pose throughout execution.

TABLE I
IN-DOMAIN AND OUT-OF-DOMAIN SIMULATION EVALUATION

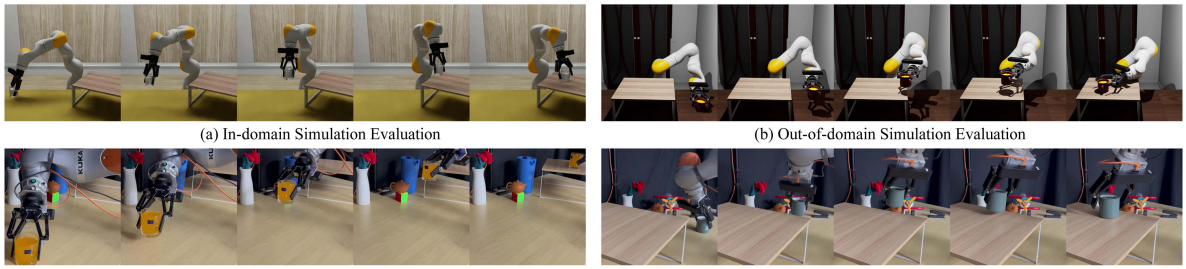
Method	SR ↑	OE ↓	MTA ↓	TS ↓
In-Domain Evaluation				
$\pi_{0.5}$ (few-shot)	11.13±0.52	1.33±0.33	18.53±4.22	277.29±115.65
$\pi_{0.5}$ (full data)	49.03±4.75	1.23±0.10	17.29±0.35	272.43±20.24
RVT-2 + DP	69.74±9.71	0.62±0.06	9.71±0.63	80.19±6.48
RVT-2 + LDP (Ours)	**80.23**±0.55	**0.55**±0.03	**8.87**±0.29	**61.98**±4.10
Out-of-Domain Evaluation (Object)				
$\pi_{0.5}$ (few-shot)	5.33±4.62	0.96±0.04	16.51±1.12	155.45±10.76
$\pi_{0.5}$ (full data)	41.60±1.46	0.91±0.07	15.34±0.62	207.92±57.91
RVT-2 + DP	69.45±7.60	0.80±0.09	11.52±1.07	83.02±8.89
RVT-2 + LDP (Ours)	**76.42**±0.29	**0.67**±0.02	**9.57**±0.11	**68.35**±5.78
Out-of-Domain Evaluation (Scene)				
$\pi_{0.5}$ (few-shot)	2.97±2.57	2.02±0.69	23.12±2.57	226.48±40.72
$\pi_{0.5}$ (full data)	30.22±6.40	1.28±0.17	18.83±0.99	178.30±21.34
RVT-2 + DP	71.79±8.58	0.75±0.14	9.48±0.77	86.23±12.98
RVT-2 + LDP (Ours)	**73.30**±4.41	**0.61**±0.05	**8.34**±0.38	**65.13**±5.99
Out-of-Domain Evaluation (Object + Scene)				
$\pi_{0.5}$ (few-shot)	9.94±1.21	**0.61**±0.48	11.92±5.48	191.61±5.07
$\pi_{0.5}$ (full data)	27.58±7.74	1.03±0.11	16.07±0.82	147.21±9.42
RVT-2 + DP	48.42±1.56	0.74±0.07	10.04±0.84	85.03±15.14
RVT-2 + LDP (Ours)	**70.08**±10.83	0.63±0.11	**8.58**±1.03	**58.13**±5.01

Out-of-domain Evaluation: As shown in Table I, our method consistently achieves the highest success rate across all OOD splits, indicating strong generalization of the learned stability-aware behavior. This advantage becomes most pronounced in the challenging OOD-object+scene setting, where our method improves upon RVT-2 + DP by 21.66% in success rate. The stability metrics also show consistent gains. Compared with RVT-2 + DP, our method generally achieves lower OE, MTA, and TS, suggesting that latent-space action generation provides more robust low-level control under distribution shifts. In contrast, the VLA baselines exhibit limited OOD robustness, especially under the compound shift, indicating that general VLA priors alone are insufficient for stable manipulation. Qualitative results of our policy in the attached video and Fig. 4(b) illustrate its robust performance under distribution shifts, showcasing stable transport with minimized tilt and smooth motion.

C. Real-World Evaluation

We further evaluate our framework on a physical robotic platform to examine its zero-shot sim-to-real transfer capability. The platform consists of a KUKA LBR iiwa 14 R820 robot arm equipped with a Robotiq 2F-140 gripper. For perception, we

Authorized licensed use limited to: Tsinghua University. Downloaded on October 19,2026 at 12:54:23 UTC from IEEE Xplore. Restrictions apply.



(a) In-domain Simulation Evaluation (b) Out-of-domain Simulation Evaluation

(c) Zero-shot Real-world Evaluation

Fig. 4. Qualitative results in simulation and the real world. (a) In-domain simulation evaluation on seen distributions. (b) Out-of-domain simulation evaluation on unseen objects and scene configurations. (c) Zero-shot real-world evaluation, demonstrating the sim-to-real transfer of the learned policy.

TABLE II
REAL-WORLD EVALUATION

Method	SR ↑	OE ↓	MTA ↓	TS ↓
$\pi_{0.5}$ (full data)	36.67	0.98	10.56	150.58
RVT-2 + DP	50.00	0.70	5.88	63.46
RVT-2 + LDP (Ours)	**73.33**	**0.65**	**3.96**	**50.73**

TABLE III
ABLATION STUDIES

Component	Variant	SR ↑	OE ↓	MTA ↓	TS ↓
Full model	–	**70.08**$\pm$10.83	0.63$\pm$0.11	**8.58**$\pm$1.03	**58.13**$\pm$5.01
Data pipeline	w/o Traj. Filter	54.02$\pm$11.08	0.72$\pm$0.03	10.07$\pm$0.34	59.42$\pm$6.47
Hierarchy	LDP-only	29.71$\pm$9.99	0.89$\pm$0.08	10.77$\pm$1.23	158.89$\pm$38.10
History	Hist-4	55.48$\pm$3.91	0.67$\pm$0.15	9.14$\pm$0.92	64.14$\pm$3.94
	Hist-12	57.40$\pm$11.45	**0.61**$\pm$0.04	8.90$\pm$0.77	58.78$\pm$5.27
Aggregation	w/o Ensemble	18.37$\pm$1.06	2.11$\pm$0.37	16.38$\pm$1.28	218.47$\pm$47.71

utilize two ZED 2i RGB-D cameras: a front camera observing the workspace and a wrist camera.

During real-world deployment, we adopt a latency-aware asynchronous execution strategy [34] to avoid inference-induced pauses, which are particularly harmful for liquid transport. While the robot executes the current action chunk, the policy infers the next chunk from the latest observations. Once the new chunk is available, actions corresponding to timesteps already elapsed during inference are discarded, and the remaining future actions are temporally ensembled with overlapping predictions for smooth continuous execution.

We evaluate the policies on a diverse set of real-world cups made of glass, plastic, and metal, which vary significantly in shape, size, reflectance, and visual appearance. Each policy is evaluated over 30 real-world trials. We omit $\pi_{0.5}$ **(few-shot)** from the real-world evaluation because its preliminary rollouts were highly unstable and unsafe for liquid transport. Crucially, all reported policies are trained purely on synthetic demonstrations, without any real-world demonstrations, fine-tuning, or adaptation. Thus, the real-world evaluation serves as a direct test of both the quality of our stability-aware synthetic data and the sim-to-real generalization of the learned policy. Qualitative real-world rollouts of our policy are provided in the attached video and Fig. 4(c).

As shown in Table II, our method achieves the best overall real-world performance among all compared methods. Since all policies are deployed without real-world adaptation, the results indicate that our simulation pipeline generates demonstrations that capture transferable stability-aware behaviors. Compared with RVT-2 + DP, our method achieves better task success, stability, and smoothness, demonstrating the advantage of the latent-space action generation over raw-action diffusion for high-frequency real-world execution. The gap over $\pi_{0.5}$ further suggests that dynamically sensitive manipulation requires more than generalist semantic and visual priors; it demands an architecture capable of generating temporally coherent, stability-aware actions. Overall, the real-world results demonstrate the zero-shot sim-to-real capability of our hierarchical policy and validate the effectiveness of training purely on synthetic, stability-aware demonstrations.

D. Ablation Study

We evaluate the contribution of each key component under the OOD-object+scene setting, as reported in Table III.

Stability-aware data filtering: Removing both tilt screening and fluid simulation (**w/o Traj. Filter**) degrades both success rate and stability. This indicates that geometric and kinematic feasibility alone is insufficient for stable dynamics, underscoring the necessity of trajectory-level dynamic validation.

Hierarchical keyframe prediction: Removing the high-level keyframe policy (**LDP-only**) causes a substantial performance drop. This confirms that the low-level policy alone struggles to infer both task-level goals and smooth executable trajectories under compound shifts, justifying the decomposition of semantic reasoning and motion generation.

Proprioceptive history length: Shortening proprioceptive history (**Hist-4**, **Hist-12**) reduces success rates compared to our full **Hist-20** model. While **Hist-12** maintains comparable stability, **Hist-20** achieves the strongest overall performance, demonstrating that longer history provides crucial temporal context for execution under distribution shifts.

Temporal ensemble: Removing temporal ensembling (**w/o Ensemble**) causes the most significant performance degradation. This component is critical for stable closed-loop execution as it suppresses abrupt chunk-boundary transitions, thereby reducing spill-inducing motions under unseen configurations.

VI. CONCLUSION

In this work, we present a synthetic-to-real framework for the dynamically stable and smooth manipulation of liquid-filled containers. Our approach combines a stability-aware synthetic data pipeline with a hierarchical latent diffusion policy, enabling the generation of smooth trajectories. Evaluations show that

Authorized licensed use limited to: Tsinghua University. Downloaded on October 19,2026 at 12:54:23 UTC from IEEE Xplore. Restrictions apply.

our framework improves success rates, trajectory stability, and motion smoothness in both simulation and real world compared to representative baselines. Notably, real-world experiments demonstrate zero-shot sim-to-real generalization without any real-world demonstrations. Future work will extend this framework to other dynamically sensitive manipulation tasks beyond liquid transport.

REFERENCES

[1] A. Goyal, V. Blukis, J. Xu, Y. Guo, Y.-W. Chao, and D. Fox, "RVT-2: Learning precise manipulation from few demonstrations," in *Proc. Robot.: Sci. Syst.*, Delft, The Netherlands, Jul. 2024, doi: 10.15607/RSS.2024.XX.055.

[2] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," in *Proc. Robot.: Sci. Syst.*, Daegu, Republic of Korea, Jul. 2023.

[3] C. Chi et al., "Diffusion policy: Visuomotor policy learning via action diffusion," *Int. J. Robot. Res.*, vol. 44, no. 10–11, pp. 1684–1704, 2025, doi: 10.15607/RSS.2023.XIX.016.

[4] M. J. Kim et al., "OpenVLA: An open-source vision-language-action model," in *Proc. 8th Conf. Robot Learn.*, Nov. 2025, vol. 270, pp. 2679–2713.

[5] K. Black et al., "π_0: A vision-language-action flow model for general robot control," in *Proc. Robot.: Sci. Syst.*, LosAngeles, CA, USA, Jun. 2025, doi: 10.15607/RSS.2025.XXI.010.

[6] J. Bjorck et al., "GR00T N1: An open foundation model for generalist humanoid robots," 2025, *arXiv:2503.14734*.

[7] K. Black et al., "$\pi_{0.5}$: A vision-language-action model with open-world generalization," in *Proc. 9th Conf. Robot Learn.*, in ser. Proceedings of Machine Learning Research, vol. 305, Sep. 2025, pp. 17–40.

[8] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "RLBench: The robot learning benchmark & learning environment," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 3019–3026, Apr. 2020.

[9] Y. Mu et al., "RoboTwin: Dual-arm robot benchmark with generative digital twins," in *Proc. Comput. Vis. Pattern Recognit. Conf.*, Jun. 2025, pp. 27649–27660.

[10] T. Chen et al., "RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation," in *Proc. 43rd Int. Conf. Mach. Learn.*, 2026.

[11] T. Mu et al., "Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations," in *Proc. Neural Inf. Process. Syst. Track Datasets Benchmarks*, vol. 1, 2021.

[12] J. Gu et al., "ManiSkill2: A unified benchmark for generalizable manipulation skills," in *Proc. Int. Conf. Learn. Representations*, 2023.

[13] S. Tao et al., "Demonstrating GPU parallelized robot simulation and rendering for generalizable embodied AI with ManiSkill3," in *Proc. Robot.: Sci. Syst.*, 2025, doi: 10.15607/RSS.2025.XXI.021.

[14] H. Xue et al., "Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation," in *Proc. Robot.: Sci. Syst.*, Los Angeles, CA, USA, Jun. 2025, doi: 10.15607/RSS.2025.XXI.052.

[15] E. Xiang, H. Ma, X. Ma, Z. Liu, and D. Huang, "GraspLDP: Towards generalizable grasping policy via latent diffusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2026, pp. 28032–28041.

[16] M. Clémente et al., "Two-steps diffusion policy for robotic manipulation via genetic denoising," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 38, 2025, pp. 139962–139984, doi: 10.52202/085713-4676.

[17] Z. Gu, M. Yang, D. Zou, and D. Xu, "Learning diffusion policy from primitive skills for robot manipulation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 40, no. 22, 2026, pp. 18262–18270, doi: 10.1609/aaai.v40i22.38889.

[18] M. Mittal et al., "Isaac Lab: A GPU-accelerated simulation framework for multi-modal robot learning," 2025, *arXiv:2511.04831*.

[19] J. Feddema, C. Dohrmann, G. Parker, R. Robinett, V. Romero, and D. Schmitt, "Robotically controlled slosh-free motion of an open container of liquid," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 1, 1996, pp. 596–602.

[20] L. Moriello, L. Biagiotti, C. Melchiorri, and A. Paoli, "Manipulating liquids with robots: A sloshing-free solution," *Control Eng. Pract.*, vol. 78, pp. 129–141, 2018.

[21] W. Aribowo, T. Yamashita, and K. Terashima, "Integrated trajectory planning and sloshing suppression for three-dimensional motion of liquid container transfer robot arm," *J. Robot.*, vol. 2015, no. 1, 2015, Art. no. 279460.

[22] B. Naseri Soufiani and M. A. Adli, "An expanded impedance control scheme for slosh-free liquid transfer by a dual-arm cooperative robot," *J. Vib. Control*, vol. 27, no. 23–24, pp. 2793–2806, 2021.

[23] A. Ferrari, R. Di Leva, S. Soprani, L. Biagiotti, G. Palli, and M. Carricato, "Time-optimal anti-sloshing trajectory planning for multiple liquid-filled containers subject to SCARA motion," *IEEE Robot. Automat. Lett.*, vol. 11, no. 2, pp. 1762–1769, Feb. 2026.

[24] K. Riou, K. Dong, K. Subrin, Y. Sun, and P. Le Callet, "From temporal-evolving to spatial-fixing: A keypoints-based learning paradigm for visual robotic manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023, pp. 1728–1734.

[25] X. Ma, S. Patidar, I. Haughton, and S. James, "Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 18081–18090.

[26] H. F. Yu and A. Altahhan, "Hierarchical learning for closed-loop robotic manipulation in cluttered scenes via depth vision, reinforcement learning, and behaviour cloning," *Electronics*, vol. 14, no. 15, 2025, Art. no. 3074.

[27] D. Wang, C. Liu, F. Chang, and Y. Xu, "Hierarchical diffusion policy: Manipulation trajectory generation via contact guidance," *IEEE Trans. Robot.*, vol. 41, pp. 2086–2104, 2025.

[28] B. Sundaralingam et al., "CuRobo: Parallelized collision-free robot motion generation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 8112–8119.

[29] H.-S. Fang et al., "AnyGrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Trans. Robot.*, vol. 39, no. 5, pp. 3929–3945, Oct. 2023.

[30] S. Bai et al., "Qwen3-VL technical report," 2025, *arXiv:2511.21631*.

[31] M. Macklin and M. Müller, "Position based fluids," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 1–12, 2013.

[32] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 6840–6851.

[33] M. Deitke et al., "Objaverse: A universe of annotated 3D objects," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 13142–13153.

[34] H. Xie et al., "DynamicVLA: A vision-language-action model for dynamic object manipulation," 2026, *arXiv:2601.22153*.

Authorized licensed use limited to: Tsinghua University. Downloaded on October 19,2026 at 12:54:23 UTC from IEEE Xplore. Restrictions apply.