QuaMa: Extending Skill Learning with Delta Dynamics Modeling for Precise and Generalizable Quantitative Manipulation

Yiwen Hou[1*], Yaoyu Cheng[1*], Jiayu Luo[1], Zhixuan Xu[1], Chongkai Gao[1], Jingxiang Guo[1], Lin Shao[1†]

***Abstract*— Quantitative manipulation requires robots not only to perform actions but also to achieve precise target quantities, a capability essential for applications such as cooking, laboratory automation, and industrial dispensing. Achieving both precision and generalization, however, remains challenging due to nonlinear and delayed dynamics, while existing end-to-end policies can be brittle under such variability. We introduce QuaMa, a framework that extends skill learning with delta dynamics modeling: a weighted behavior cloning policy captures general skills, while a delta-based Quantity Prediction Model (QPM) forecasts future quantity changes under candidate actions. During execution, Model Predictive Action Refinement (MPAR) leverages QPM to refine the skill policy's output, ensuring accurate and reliable control. This modular design explicitly reasons about quantities while retaining compatibility with advances in imitation learning. We validate QuaMa on real-world pouring tasks with diverse containers, liquids, and granules. Results show that QuaMa consistently achieves high accuracy and robust zero-shot generalization to unseen containers, materials, and target quantities. The overall mean absolute error across all evaluated conditions remains below 1 g, demonstrating both precision and generalization. Project page: `https://quantitativemanipulation.github.io/`**

I. Introduction

Robotic manipulation has achieved remarkable progress in recent years, enabling robots to perform tasks ranging from grasping to assembly [1]–[6]. However, most existing studies remain focused on skill learning [7]–[10], emphasizing how to imitate or execute action sequences, while the crucial ability of quantitative manipulation has received little attention. Yet quantitative manipulation is indispensable in many real-world applications. In food preparation [11], [12], robots must dispense precise amounts of condiments, grains, or liquids to ensure taste and nutrition. In laboratory automation [13], [14], accurate dosing of chemicals is critical for reliable experiments and drug development. These scenarios demand that robots not only perform the skill-level actions but also achieve precise and generalizable quantitative manipulation.

Achieving such capabilities is particularly challenging. The relationship between robot actions and the resulting quantities is highly complex: the same action sequences can produce different outcomes depending on container geometry or material properties. The system further exhibits nonlinear, delayed, and irreversible dynamics, as once liquid or granular material is poured, it cannot be retrieved.

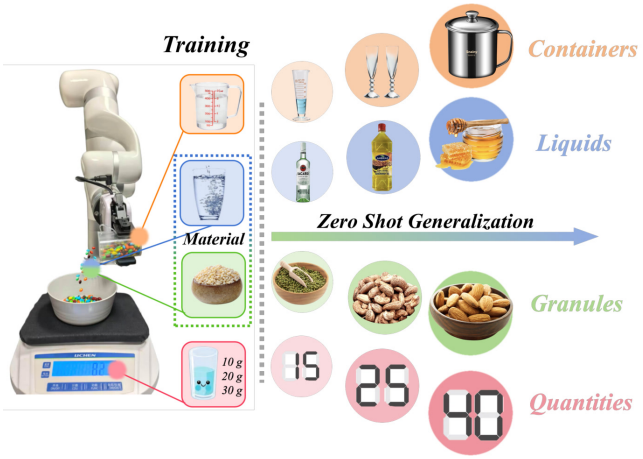

Fig. 1. Trained on a small set of real-world demonstrations, our method enables zero-shot generalization to unseen containers, liquids, granules, and target quantities.

End-to-end imitation learning—especially behavior cloning (BC)—has shown strong capability in acquiring complex skills from demonstrations [7]–[10]. However, BC chiefly optimizes action similarity rather than the quantities achieved. Under the nonlinear, delayed, and partly irreversible dynamics of quantitative manipulation (e.g., pouring), small action errors can compound: the same motion can induce different quantity increments depending on container geometry and material flow, and dispensed mass cannot be recovered. As a result, policies that imitate at the trajectory level can still miss the target quantity, with errors exacerbated when generalizing to unseen containers, materials, or target quantities.

To address these challenges, we propose QuaMa, a framework that extends skill learning with quantity prediction. A weighted BC policy first learns feasible skill trajectories through imitation. On top of this, a delta-based Quantity Prediction Model (QPM) forecasts future quantity changes under candidate actions. During execution, Model Predictive Action Refinement (MPAR) leverages QPM to refine the policy outputs, explicitly aligning skill execution with quantitative objectives and dynamically adjusting actions to achieve precise outcomes across diverse conditions.

While effective, the framework poses two key challenges, each tied to a different module. First, training the Quantity Prediction Model (QPM) to predict absolute quantities

This research is supported by the Ministry of Education, Singapore, under the Academic Research Fund Tier 1 (FY2024).

*Equal contribution. †Corresponding author.

[1]Yiwen Hou, Yaoyu Cheng, Jiayu Luo, Zhixuan Xu, Chongkai Gao, Jingxiang Guo, and Lin Shao are with the National University of Singapore. `yiwenhou@u.nus.edu, linshao@nus.edu.sg`

yields a dispersed target distribution and weak extrapolation. We therefore reformulate QPM as a delta dynamics model, expressing predictions as relative changes with respect to the current state; this compact parameterization improves in-distribution accuracy and naturally enables out-of-distribution generalization. Second, the BC module faces severe phase imbalance: decisive near-goal "stopping" states are rare yet critical, whereas early-stage states dominate the data. We address this with a goal-aware weighting scheme applied during BC training, up-weighting near-goal samples to steer the learned skill policy toward precise stopping, thus reinforcing the overall framework's ability to achieve accurate quantitative control.

We validate QuaMa on a real-world robotic platform using two representative categories of quantitative manipulation as testbeds: pouring diverse liquids (e.g., water, oil, honey, and milk) and pouring diverse granules (e.g., coix seeds, red beans, and broad beans). Although both testbeds involve pouring, they exhibit distinct continuous-flow and discrete-particle dynamics, while the QuaMa formulation applies more broadly to quantity-conditioned manipulation. Experiments demonstrate improved quantitative accuracy and zero-shot generalization to unseen containers, materials, and target quantities within the evaluated categories.

In conclusion, our primary contributions are as follows:

1) We introduce and formalize the problem of quantitative manipulation, and present QuaMa, a framework that extends skill learning with quantity prediction, establishing a foundation for studying precise and generalizable quantitative manipulation.
2) We introduce two key innovations: a delta dynamics model that enhances prediction accuracy and extrapolation, and a goal-aware weighting strategy that mitigates critical data imbalance, enabling accurate control under complex real-world dynamics.
3) We validate QuaMa on real robots with diverse liquids and granules, demonstrating significant improvements over strong baselines and showcasing its effectiveness and robustness in real-world quantitative manipulation.

II. Related Work

A. Robotic Pouring and Quantitative Manipulation

Robotic pouring has been studied through perception, feedback control, and learning. Related methods improve visual perception, spatial reasoning, or general skill generation for successful task execution [10], [15]–[17]. For quantitative pouring, visual closed-loop approaches estimate liquid levels or flow and regulate the pouring motion or stopping time [18]–[20]. Although effective in controlled settings, feedback based only on the current observation can react late to delayed and irreversible flow. Multimodal methods improve flow estimation with audio and haptics, while temporal policies use observation histories to capture uncertain liquid dynamics [21]–[23]. Simulation-based reinforcement learning and action optimization offer another route, and have demonstrated transfer to real liquid pouring and powder weighing [24]–[26]; however, they typically rely on task-specific simulation, dynamics randomization, or material calibration.

Across these directions, quantity regulation is often implicit in an end-to-end policy, hand-designed feedback, or a task-specific model. QuaMa instead separates skill execution from quantity regulation and learns quantity-change dynamics directly from real feedback, without explicit physical simulation or material calibration.

B. Learning for Manipulation under Complex Dynamics

Beyond pouring-specific studies, imitation learning has acquired complex manipulation skills with temporally expressive policies, including action-chunking Transformers and diffusion models [7], [8], [10]. Recent vision-language-action models further show broad task generalization [9], [27], [28]. Nevertheless, BC optimizes action imitation rather than the final task quantity; even a sequence policy can therefore reproduce plausible motion while accumulating an irreversible quantity error.

RL fine-tuning can improve precision but requires additional interaction [2], [29], [30]. Model-based methods instead support planning through learned or approximate dynamics [31], [32], but full-state physical prediction is difficult for fluids and granules. QuaMa does not replace the underlying sequence policy: it reweights critical demonstrations and adds a compact, history-conditioned quantity-change model for online refinement.

Compared with pure trajectory imitation or explicit physical simulation, our focus is quantity-level delta dynamics and direct optimization of the target quantity from real feedback.

III. Problem Formulation

We study the problem of *quantitative manipulation*, where a robot must not only execute a manipulation skill (e.g., pouring) but also achieve a precise target quantity. This setting extends traditional manipulation tasks, which are typically evaluated by binary success or failure, by explicitly incorporating the notion of quantity.

Formally, let $s_t \in \mathcal{S}$ denote the observable state of the robot and environment at time t. The state includes the robot's proprioceptive and perceptual information, together with the current task quantity $q_t \in \mathcal{Q}$. The robot executes actions $a_t \in \mathcal{A}$, which influence the evolution of q_t.

Each task is specified by a target quantity $q^* \in \mathcal{Q}$. The policy is defined as:

$$\pi(a_t \mid s_t, q^*).$$

The objective is to minimize the expected target deviation:

$$\min_{\pi} \ \mathbb{E}_{q^* \sim \mathcal{Q}}[\,|q_T - q^*|\,], \tag{1}$$

where q_T denotes the quantity at the end of execution.

This formulation highlights two central challenges. First, in many quantitative manipulation tasks (e.g., pouring), the mapping from actions to quantities is nonlinear, delayed, and irreversible, which makes it difficult to precisely anticipate the outcome of actions, especially when approaching the

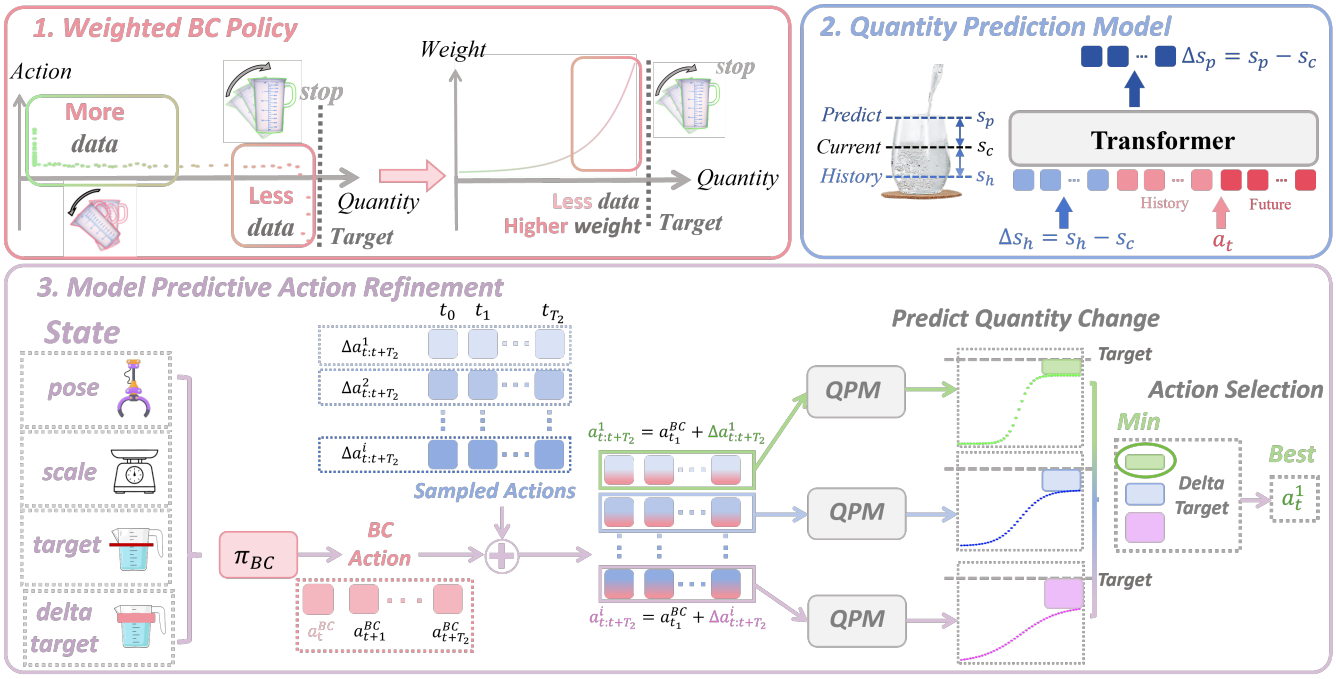

Fig. 2. Overview of QuaMa. (1) A weighted BC policy addresses data imbalance by emphasizing actions near the target quantity. (2) A delta-based Quantity Prediction Model (QPM) predicts future quantity changes from history and candidate actions. (3) During execution, Model Predictive Action Refinement (MPAR) combines BC outputs with QPM predictions to refine actions online, enabling precise and generalizable quantitative manipulation.

target. Second, the robot must generalize across diverse containers, materials, and target quantities.

IV. Method

We present QuaMa, a framework for precise and generalizable quantitative manipulation that *extends skill learning with delta dynamics modeling*. As shown in Fig. 2, QuaMa is composed of three modules: (i) a weighted behavior cloning policy that learns general manipulation skills while emphasizing rare but critical near-target actions (Section IV-B); (ii) a delta-based Quantity Prediction Model (QPM) that predicts how candidate actions influence future quantities (Section IV-C); and (iii) Model Predictive Action Refinement (MPAR), which leverages QPM to refine policy outputs online and drive the system toward the target quantity (Section IV-D). This modular design reduces learning complexity, explicitly aligns with quantitative objectives, and enables robust performance across diverse containers, materials, and target quantities.

A. Decoupling Skill and Quantity

Behavior cloning (BC) is a widely used and effective approach for learning complex manipulation skills from demonstrations. However, in quantitative manipulation, a central challenge arises from the mismatch between action-level imitation and quantity-level accuracy. BC minimizes the discrepancy between predicted and demonstrated actions,

$$\mathcal{L}_{\text{BC}}(\theta) = \mathbb{E}_{(s_t, q^*, a_t^{\text{exp}})}\left[\|a_t - a_t^{\text{exp}}\|^2\right], \tag{2}$$

whereas the objective is to minimize the final quantity error,

$$\min_{\pi} \ \mathbb{E}\left[|q_T - q^*|\right]. \tag{3}$$

As a result, a policy trained solely with BC may reproduce expert trajectories without reliably achieving the desired target quantity, particularly under distribution shifts.

QuaMa addresses this challenge with a *decoupled design*. The skill policy generates feasible action proposals, while a separate Quantity Prediction Model (QPM) evaluates their quantitative effects. During execution, the two components are integrated via Model Predictive Action Refinement (MPAR): the skill policy provides a base action, and MPAR refines it using QPM predictions to minimize deviation from q^*. This shifts the optimization focus from trajectory imitation to explicit quantity prediction, thereby bridging the gap between BC training and quantity-level accuracy. Details of the refinement procedure are provided in Section IV-D.

B. Skill Policy with Weighted Behavior Cloning

The skill policy of QuaMa is based on BC from demonstrations. Standard BC optimizes the policy $\pi_\theta(a_t \mid s_t, q^*)$ with the objective defined in Eq. (2).

However, applying vanilla BC directly to quantitative manipulation reveals a key challenge: the demonstration data are highly imbalanced. Most trajectories correspond to early-stage pouring, while near-target actions (e.g., stopping at the right moment) are both rare and critical. This imbalance is illustrated in Fig. 2, where decisive stopping actions are underrepresented compared to early pouring states. As a result, policies trained with vanilla BC tend to underfit these crucial states, leading to inaccurate actions and weaker initializations for subsequent optimization.

To overcome this limitation, we introduce a *quantity-aware weighting scheme*. Each sample is weighted according

to the distance between the current quantity q_t and the target quantity q^*:

$$w_t = \exp\big(-\alpha\,|q_t - q^*|\big), \tag{4}$$

where $\alpha > 0$ is a scaling factor. This emphasizes states closer to the target, encouraging the policy to reproduce rare but decisive stopping actions while still learning from earlier stages. The resulting weighted BC objective is

$$\mathcal{L}_{\text{wBC}}(\theta) = \mathbb{E}_{(s_t, q^*, a_t^{\text{exp}})}\big[w_t\,\|a_t - a_t^{\text{exp}}\|^2\big]. \tag{5}$$

With this scheme, the skill policy generates more accurate actions in the critical near-target region, offering stronger proposals for refinement.

C. Quantity Prediction Model (QPM)

The second component of QuaMa is the *Quantity Prediction Model (QPM)*, which predicts how future quantities evolve under candidate action sequences. As shown in Fig. 2, QPM takes as input a history of observed quantities $(q_{t-H+1}, \ldots, q_t)$, a history of executed actions $(a_{t-H}, \ldots, a_{t-1})$, and a candidate sequence of future actions $(a_t, \ldots, a_{t+K-1})$. It then predicts the corresponding future quantities $(\hat{q}_{t+1}, \ldots, \hat{q}_{t+K})$. We implement QPM with a Transformer, which is well-suited for modeling temporal dependencies across past and future sequences.

A straightforward approach is to directly predict *absolute quantities*:

$$\hat{q}_{t+k} = f_\phi(q_{t-H+1:t}, a_{t-H:t+k-1}), \quad k = 1, \ldots, K, \tag{6}$$

but this suffers from two drawbacks. First, absolute quantities grow monotonically in tasks such as pouring, so samples from different stages are spread out in prediction space, leading to inefficient learning. Second, absolute prediction restricts extrapolation, since the model is trained only within the target range present in demonstrations.

To address these issues, we adopt a *delta dynamics formulation*. All quantities are normalized with respect to the current q_t, and the model predicts future changes rather than absolute values:

$$\begin{aligned} \Delta q_{t-i} &= q_{t-i} - q_t, \quad i = 0, \ldots, H-1, \\ \Delta\hat{q}_{t+k}(\phi) &= f_\phi(\Delta q_{t-H+1:t}, a_{t-H:t-1}, a_{t:t+k-1}), \\ \hat{q}_{t+k} &= q_t + \Delta\hat{q}_{t+k}(\phi), \quad k = 1, \ldots, K. \end{aligned} \tag{7}$$

This representation aligns trajectories from different stages into a compact space. As illustrated in Fig. 3, absolute-dynamics predictions remain dispersed across different trajectory stages, while delta dynamics maps these stages into a compact space, yielding a more concentrated distribution. This compactness makes the prediction problem easier and naturally supports extrapolation to unseen target quantities.

The QPM is trained with a regression loss:

$$\mathcal{L}_{\text{QPM}}(\phi) = \frac{1}{NK}\sum_{t=1}^{N}\sum_{k=1}^{K}\left\|(q_{t+k} - q_t) - \Delta\hat{q}_{t+k}(\phi)\right\|^2. \tag{8}$$

By explicitly modeling action-to-quantity dynamics, QPM captures the delayed and nonlinear effects of actions, while the delta formulation yields a compact space that generalizes across diverse containers, materials, and target quantities.

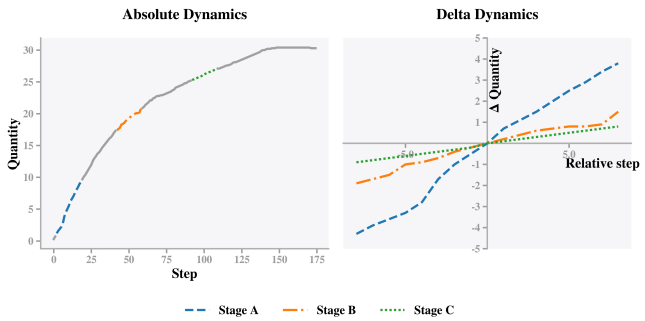

Fig. 3. Comparison of absolute (left) and delta (right) dynamics. Delta dynamics modeling maps different stages into a compact space, which simplifies learning and supports extrapolation to unseen target quantities.

D. Model-Predictive Action Refinement (MPAR)

The final component of QuaMa is the *Model-Predictive Action Refinement (MPAR)* module, illustrated in Fig. 2. At each step t, the skill policy outputs a candidate action sequence $(a_t^{\text{BC}}, \ldots, a_{t+K-1}^{\text{BC}})$. To refine this sequence, we sample a set of M adjustment sequences:

$$\mathcal{D} = \{(\Delta a_t^{(j)}, \ldots, \Delta a_{t+K-1}^{(j)})\}_{j=1}^{M}. \tag{9}$$

Each adjustment sequence is added to the skill policy's output to form a refined candidate:

$$\mathbf{a}^{(j)} = (a_t^{\text{BC}} + \Delta a_t^{(j)}, \ldots, a_{t+K-1}^{\text{BC}} + \Delta a_{t+K-1}^{(j)}). \tag{10}$$

for $j = 1, \ldots, M$.

For each candidate sequence $\mathbf{a}^{(j)}$, the QPM predicts the corresponding quantities:

$$(\hat{q}_{t+1}^{(j)}, \ldots, \hat{q}_{t+K}^{(j)}) = \text{QPM}(s_t, \mathbf{a}^{(j)}). \tag{11}$$

The optimal adjustment sequence is then selected by minimizing the deviation from the target:

$$j^* = \arg\min_{j \in \{1,\ldots,M\}} \left|\hat{q}_{t+K}^{(j)} - q^*\right|. \tag{12}$$

Finally, only the first action of the best refined sequence is executed:

$$a_t^* = a_t^{\text{BC}} + \Delta a_t^{(j^*)}. \tag{13}$$

The process repeats in a receding-horizon manner: after each action, the system receives new observations, updates QPM's input, and performs another round of refinement. This rolling correction allows the robot to adapt online to prediction errors and environmental variations.

Through MPAR, BC ensures feasible skill execution, while QPM-based corrections explicitly guide execution toward the target quantity. This resolves the mismatch between action-level imitation and quantity-level objectives, and leverages the compact delta space of QPM to generalize across containers, materials, and target quantities. Taken together, the three components of QuaMa provide a unified framework for quantitative manipulation: weighted BC learns feasible skills, QPM models action-to-quantity dynamics in a compact delta space, and MPAR closes the loop by refining the skill policy's actions online. This integration enables QuaMa to

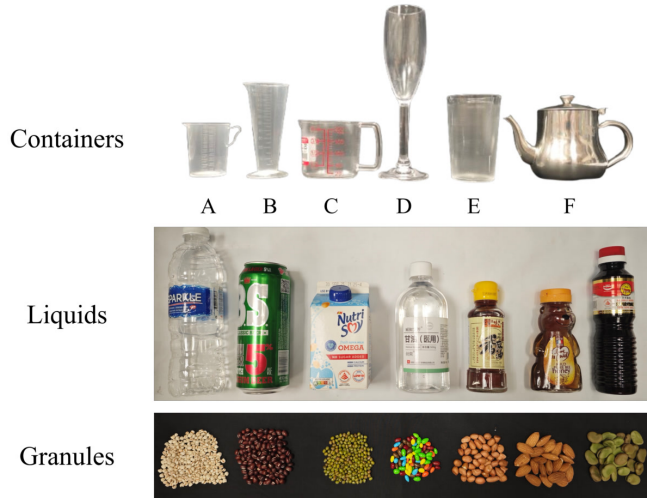

Fig. 4. Containers, liquids, and granules used in the real-world experiments.

reproduce expert skills while achieving precise and reliable quantitative control under real-world uncertainties.

V. Experiments

In this section, we perform a series of experiments aimed at addressing the following questions (Q1-Q5):

Q1: Can QuaMa achieve precise quantitative manipulation in real-world tasks?
Q2: Does QuaMa generalize effectively to *unseen containers*, *unseen materials*, and *unseen target quantities*?
Q3: How does the QPM with MPAR refine the actions from the skill policy?
Q4: What is the impact of using *delta dynamics* compared to absolute dynamics in QPM?
Q5: How does the *goal-aware weighting strategy* affect the performance of the skill learning module?

A. Experimental Setup

Platform and Tasks. All experiments are conducted on an xArm manipulator equipped with a high-precision digital scale that provides real-time feedback on the poured quantity. The control loop runs at 10 Hz. We evaluate pouring **liquids** and **granules** from different **containers**. Figure 4 shows the experimental objects. The top row contains six source containers of varying shapes and capacities. The middle row shows the liquids, from left to right: water, beer, milk, glycerol, oil, honey, and dark soya sauce. The bottom row shows the granules, from left to right: Job's tears (coix), red beans, green beans, candy-coated chocolate, peanuts, almonds, and broad beans. These quantitative pouring tasks are designed to study generalization along three axes: across different containers, across different liquids and granules, and across different target quantities. Each trial requires the robot to pour a specified target quantity into the receiving container, thereby covering both continuous fluid flow and discrete particle dynamics.

Data Collection. Based on the above setup, we collect expert demonstrations via teleoperation using a Meta Quest 3 headset and controllers. At the start of each trajectory, the robot grasps a source container filled with a selected liquid or granules and holds it upright above a target container. Each demonstration starts with the container upright, then tilts to pour, stops once the target quantity is reached, and returns upright.

For data collection, we use **Container A** together with two liquids (*water*, *honey*) and one type of granules (*coix*). For each material, we record 90 trajectories covering 3 target quantities (10, 20, and 30 g), with 30 repetitions per target and varied initial fill levels. All other containers and the remaining liquids and granules are reserved exclusively for **zero-shot generalization testing**, allowing us to evaluate how well the learned policy transfers across containers, materials, and target quantities not seen during training.

Evaluation Protocol and Metrics. In each trial, the robot is given a target quantity q^* and executes a pouring trajectory until termination, yielding a final poured quantity q_T. The primary metric is the absolute error at termination:

$$\text{Error} = |q_T - q^*|.$$

To provide a more complete evaluation, we additionally report:

- **Success@ τ:** proportion of trials with $|q_T - q^*| \leq \tau$.
- **Overshoot / Undershoot:** decomposition of signed error into positive and negative components:

$$\text{Over} = \tfrac{1}{N}\sum_{i=1}^{N} \max(0,\, q_T^{(i)} - q^{*(i)}), \tag{14}$$

$$\text{Under} = \tfrac{1}{N}\sum_{i=1}^{N} \max(0,\, q^{*(i)} - q_T^{(i)}). \tag{15}$$

Baselines. We compare QuaMa against two representative methods:

- **PD control:** following prior work [18], a controller that regulates the 1D end-effector rotation angle based on the error between the current poured quantity and the target, and automatically stops once the target quantity is reached.
- **Behavior Cloning (BC):** a policy that directly imitates expert demonstrations, using ACT for liquids (smooth continuous dynamics) and Diffusion Policy for granules (multimodal discrete dynamics).

These baselines represent heuristic control and pure imitation learning, providing complementary points of comparison for evaluating QuaMa.

Implementation Details. The BC models take the current observation and predict an action chunk of $K = 8$ complete 6D end-effector poses; we set $\alpha = 10$. QPM uses history $H = 10$ and horizon $K = 8$. MPAR keeps the other five action dimensions fixed and refines only the end-effector roll used for tilting, exhaustively evaluating $\mathcal{A}_\Delta = \{-0.5°, 0°, 0.5°\}$ over the horizon ($M = 3^8 = 6561$). All candidates are evaluated in parallel as a single QPM batch. On an NVIDIA RTX 4090, the measured worst-case latency of model inference and action selection is below 0.1 s.

Method	Liquids			Granules		
	MAE ↓	Succ@1g ↑	Over / Under	MAE ↓	Succ@1g ↑	Over / Under
PD control	0.643	71.4%	0.643 / 0.000	7.984	0%	7.170 / 0.814
BC	0.508	87.5%	0.255 / 0.253	1.075	52.5%	0.785 / 0.290
QuaMa	**0.170**	**100%**	**0.095 / 0.075**	**0.248**	**100%**	**0.080 / 0.168**

TABLE I

IN-DISTRIBUTION (ID) RESULTS ON LIQUIDS AND GRANULES. WE REPORT MEAN ABSOLUTE ERROR (MAE, G), SUCCESS RATE WITHIN 1 G (SUCC@1G), AND OVERSHOOT/UNDERSHOOT (G).

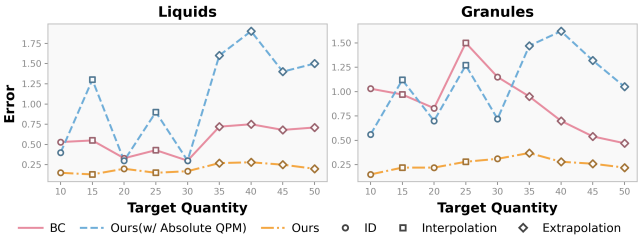

Fig. 5. OOD generalization results across target quantities.

Scalability. MPAR searches only the task-relevant roll rather than the full 6D action space. Although the number of candidates grows exponentially with the horizon, this reduced search space keeps exhaustive refinement within the 10 Hz control loop. Finer discretization or a longer horizon therefore trades computation for control granularity.

B. In-Distribution Performance

We first evaluate QuaMa under the in-distribution (ID) setting, which corresponds to **Q1**. We compare with the two baselines: PD control and BC. Table I summarizes the quantitative results across both liquids and granules.

Overall, QuaMa achieves the lowest error and the highest success rate. BC can roughly imitate expert trajectories but lacks the precision to stop exactly at the target, leading to larger deviations. PD control performs reasonably on liquids but fails on granules, whose complex dynamics require six-dimensional motions (e.g., shaking) for accurate pouring. Because PD regulates only one-dimensional rotation, it cannot execute these motions and therefore incurs large errors in granular pouring.

C. Out-of-Distribution Generalization

We next evaluate QuaMa under the out-of-distribution (OOD) setting, which addresses **Q2**. During training, only **Container A** is used together with two liquids (*water*, *honey*) and one type of granules (*coix*). The target quantities in training are limited to 10, 20, and 30 g. All other containers, liquids, granules, and target quantities are excluded from training and reserved for zero-shot testing.

We first investigate generalization across containers and materials. Table II reports the averaged final error (g) for all container–material combinations. Each entry is obtained by averaging 9 trials: 3 target quantities (20, 15, and 40 g) × 3 repetitions. These targets correspond to in-distribution (20 g), interpolation (15 g), and extrapolation (40 g). The rightmost columns show per-container averages (**L-Avg**, **G-Avg**), and the bottom row shows per-material average results.

QuaMa maintains consistently low errors across unseen containers, liquids, and granules, showing only modest degradation compared to in-distribution performance. Its performance remains stable across containers of different shapes and capacities, as well as materials with varying viscosities and particle dynamics. These results confirm that QuaMa achieves robust generalization across container geometry and material properties.

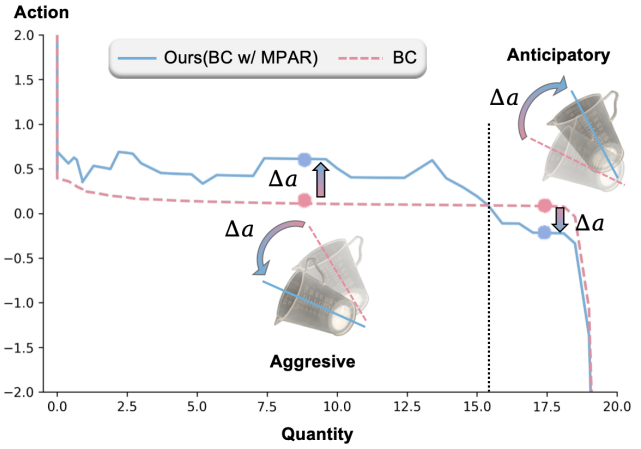

Fig. 6. Action refinement with MPAR compared to raw BC.

We further analyze generalization with respect to **target quantities**. Figure 5 reports the final error across different target quantities. 10, 20, and 30 g are in-distribution (ID); 15 and 25 g are interpolation; and 35–50 g are extrapolation.

QuaMa achieves consistently low errors across all regimes. Performance remains stable under both interpolation and extrapolation, demonstrating the ability to generalize to unseen target quantities. The ablations below attribute this robustness to the **delta QPM**.

D. Policy Refinement with QPM and MPAR

We now turn to **Q3**, which investigates how the QPM together with MPAR improves the raw actions from the skill policy. While the skill policy (BC) can imitate feasible pouring trajectories, it often lacks the precision to stop exactly at the target. QPM provides short-horizon predictions of quantity change given candidate actions, and MPAR leverages this model to refine the policy output by selecting actions that better align with the target quantity.

Figure 6 illustrates how MPAR refines the output of the skill policy. The x-axis denotes the poured quantity and the y-axis the action value, with the target set to 20 g. The raw BC policy (red) produces smoother but less precise actions, while BC with MPAR (blue) makes more aggressive choices when far from the target, allowing faster approach to the target. Near the goal region (15–20 g), MPAR anticipates the stopping condition and reduces the tilt earlier, thereby avoiding overshoot.

This refinement effect complements the quantitative results

TABLE II

OOD GENERALIZATION RESULTS OF QUAMA ACROSS CONTAINERS AND MATERIALS.

	Liquids								Granules							
Container	Water	Beer	Milk	Glycerol	Oil	Honey	Sauce	**L-Avg**	Coix	Red	Green	Candy	Peanut	Almond	Broad	**G-Avg**
	0.20	0.44	0.41	0.62	0.31	0.19	0.54	**0.39**	0.30	0.52	0.48	0.43	0.51	0.53	0.76	**0.50**
	0.34	0.21	0.30	0.79	0.72	0.53	0.53	**0.49**	0.44	0.46	0.50	0.59	0.52	0.49	0.48	**0.50**
	0.24	0.28	0.26	0.23	0.27	0.21	0.27	**0.25**	0.44	0.63	0.44	0.43	0.70	0.66	1.06	**0.62**
	0.28	0.32	0.28	0.81	0.74	0.78	0.87	**0.58**	0.61	0.73	0.36	0.57	0.87	0.93	1.28	**0.76**
	0.26	0.31	0.18	1.16	0.44	0.54	0.28	**0.45**	0.29	0.56	0.60	0.56	0.83	1.19	1.31	**0.76**
	0.41	0.54	0.64	1.01	0.47	0.91	1.31	**0.76**	-	-	-	-	-	-	-	-
Average	**0.29**	**0.35**	**0.35**	**0.77**	**0.49**	**0.53**	**0.63**	**0.49**	**0.42**	**0.58**	**0.48**	**0.52**	**0.69**	**0.76**	**0.98**	**0.63**

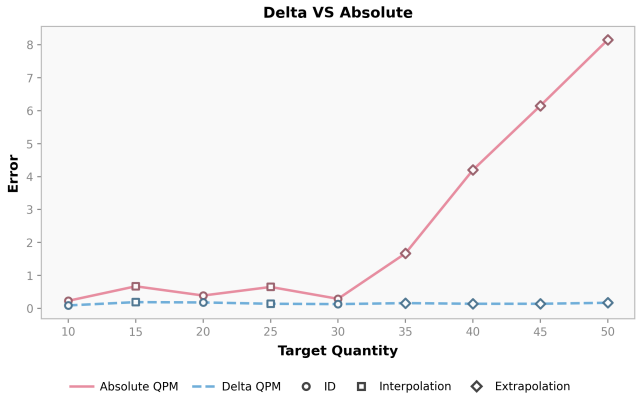

Fig. 7. Comparison of absolute and delta dynamics in QPM.

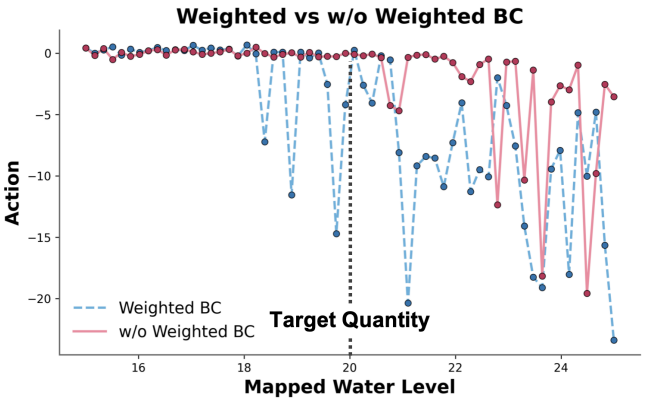

Fig. 8. Comparison of weighted vs. vanilla BC.

in Table I, where QuaMa achieves higher accuracy and stability than BC alone. Together, these findings confirm that QuaMa effectively aligns skill with target quantities.

E. Delta Dynamics vs. Absolute Dynamics

We now address **Q4**, which examines the impact of using *delta dynamics* in the Quantity Prediction Model (QPM). We compare two formulations: (1) an *absolute* model that directly predicts the future quantity given the history state and action, and (2) a *delta* model that predicts the quantity change induced by the action.

Figure 7 reports the prediction error of the two models across different target quantities. For each target (10–50 g), we run online rollouts using our method, record the trajectories, and evaluate QPM predictions with both the absolute and delta formulations. The results show that on in-distribution targets (10, 20, 30 g), both models achieve similar accuracy. For interpolation targets (15, 25 g), the absolute model begins to degrade, while the delta model maintains stable error. Most importantly, for extrapolation targets (35–50 g), the error of the absolute model grows rapidly, whereas the delta model remains robust and achieves consistently low error. This demonstrates that the delta formulation not only matches the absolute model within the training range, but also provides strong interpolation and extrapolation generalization.

These findings are consistent with the OOD quantity generalization results in Figure 5. When integrated with MPAR, the absolute model can improve over BC in in-distribution settings, but under extrapolation it fails and even harms performance. In contrast, the delta model maintains robust performance across all target quantities, highlighting its crucial role in enabling QuaMa to generalize reliably.

F. Effect of Goal-Aware Weighting Strategy

We finally address **Q5** by evaluating the impact of the goal-aware weighting strategy. Figure 8 illustrates the predicted actions for a target quantity of 20 g. Vanilla BC (red) exhibits noticeable underfitting in the near-goal region due to data imbalance, triggering the stopping action only after surpassing 22 g and consequently leading to overshoot.

In contrast, the weighted BC policy (blue) better captures the critical near-target dynamics. It anticipates the stopping point earlier, producing deceleration and stopping actions around 18 g, thereby enabling more reliable and precise regulation of the final quantity.

Reweighting near-target samples mitigates phase imbalance, improving stopping accuracy and subsequent action proposals.

To clarify what each ablation measures, weighted BC is compared with vanilla BC while keeping the policy architecture and demonstrations fixed; thus, the difference isolates goal-aware reweighting. Likewise, delta and absolute QPMs are evaluated within the same MPAR refinement pipeline, so their difference isolates the choice of dynamics representation rather than the refinement procedure.

VI. Conclusion

We presented **QuaMa**, a framework for precise and generalizable quantitative manipulation that combines goal-aware weighted BC, a delta-based Quantity Prediction Model, and MPAR. Experiments on diverse liquids and granules demonstrate lower error, higher success rates, and zero-shot generalization to unseen containers, materials, and target quantities. These results show that quantitative dynamics prediction and online refinement can improve learned skill execution. Our evaluation is limited to pouring; task-level transfer and unified quantitative/non-quantitative policies remain open. Extending QuaMa requires an appropriate quantity signal and identifiable goal-proximal samples; future work will explore task-conditioned dynamics, multitask policies, more efficient refinement, and richer sensing.

References

[1] Z. Wei, Z. Xu, J. Guo, Y. Hou, C. Gao, Z. Cai, J. Luo, and L. Shao, "$\mathcal{D}(\mathcal{R}, \mathcal{O})$ grasp: A unified representation of robot and object interaction for cross-embodiment dexterous grasping," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 4982–4988.

[2] L. Ankile, A. Simeonov, I. Shenfeld, M. Torne, and P. Agrawal, "From imitation to refinement - residual rl for precise assembly," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 01–08.

[3] C. Tie, S. Sun, J. Zhu, Y. Liu, J. Guo, Y. Hu, H. Chen, J. Chen, R. Wu, and L. Shao, "Manual2skill: Learning to read manuals and acquire robotic skills for furniture assembly using vision-language models," *arXiv preprint arXiv:2502.10090*, 2025.

[4] Z. Xu, Z. Xian, X. Lin, C. Chi, Z. Huang, C. Gan, and S. Song, "Roboninja: Learning an adaptive cutting policy for multi-material objects," *arXiv preprint arXiv:2302.11553*, 2023.

[5] C. Chi, B. Burchfiel, E. Cousineau, S. Feng, and S. Song, "Iterative residual policy: for goal-conditioned dynamic manipulation of deformable objects," *The International Journal of Robotics Research*, vol. 43, no. 4, pp. 389–404, 2024.

[6] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, K. Ghasemipour, C. Finn, and A. Wahid, "Aloha unleashed: A simple recipe for robot dexterity," *arXiv preprint arXiv:2410.13126*, 2024.

[7] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.

[8] 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," *The International Journal of Robotics Research*, p. 02783649241273668, 2023.

[9] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "π_0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.

[10] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu, "RDT-1b: a diffusion foundation model for bimanual manipulation," in *International Conference on Learning Representations (ICLR)*, 2025.

[11] J. Liu, Y. Chen, Z. Dong, S. Wang, S. Calinon, M. Li, and F. Chen, "Robot cooking with stir-fry: Bimanual non-prehensile manipulation of semi-fluid objects," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5159–5166, 2022.

[12] H. Shi, H. Xu, S. Clarke, Y. Li, and J. Wu, "Robocook: Long-horizon elasto-plastic object manipulation with diverse tools," in *Proceedings of The 7th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Tan, M. Toussaint, and K. Darvish, Eds., vol. 229. PMLR, 06–09 Nov 2023, pp. 642–660.

[13] Q. Zhu, F. Zhang, Y. Huang, H. Xiao, L. Zhao, X. Zhang, T. Song, X. Tang, X. Li, G. He *et al.*, "An all-round ai-chemist with a scientific mind," *National Science Review*, vol. 9, no. 10, p. nwac190, 2022.

[14] Z. Zhang, C. Yue, H. Xu, M. Liao, X. Qi, H.-a. Gao, Z. Wang, and H. Zhao, "Robochemist: Vision-language-action models for robotic chemistry," in *Conference on Robot Learning (CoRL)*, 2025.

[15] H. Lin, Y. Fu, and X. Xue, "Pourit!: Weakly-supervised liquid perception from a single image for visual closed-loop robotic pouring," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 241–251.

[16] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, "Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," in *8th Annual Conference on Robot Learning*, 2024.

[17] Z. Chen, Z. Ji, J. Huo, and Y. Gao, "Scar: Refining skill chaining for long-horizon robotic manipulation via dual regularization," *Advances in Neural Information Processing Systems*, vol. 37, pp. 111 679–111 714, 2024.

[18] C. Schenck and D. Fox, "Visual closed-loop control for pouring liquids," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 2629–2636.

[19] G. Narasimhan, K. Zhang, B. Eisner, X. Lin, and D. Held, "Self-supervised transparent liquid segmentation for robotic pouring," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 4555–4561.

[20] F. Zhu, S. Hu, L. Leng, A. Bartsch, A. George, and A. B. Farimani, "Pour me a drink: robotic precision pouring carbonated beverages into transparent containers," *arXiv preprint arXiv:2309.08892*, 2023.

[21] H. Liang, C. Zhou, S. Li, X. Ma, N. Hendrich, T. Gerkmann, F. Sun, M. Stoffel, and J. Zhang, "Robust robotic pouring using audition and haptics," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10 880–10 887.

[22] R. Feng, D. Hu, W. Ma, and X. Li, "Play to the score: Stage-guided dynamic multi-sensory fusion for robotic manipulation," in *Proceedings of The 8th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, P. Agrawal, O. Kroemer, and W. Burgard, Eds., vol. 270. PMLR, 06–09 Nov 2025, pp. 340–363.

[23] N. Saito, N. B. Dai, T. Ogata, H. Mori, and S. Sugano, "Real-time liquid pouring motion generation: End-to-end sensorimotor coordination for unknown liquid dynamics trained with deep neural networks," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2019, pp. 1077–1082.

[24] C. Do, C. Gordillo, and W. Burgard, "Learning to pour using deep deterministic policy gradients," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3074–3079.

[25] T. Lopez-Guevara, R. Pucci, N. K. Taylor, M. U. Gutmann, S. Ramamoorthy, and K. Subr, "Stir to pour: Efficient calibration of liquid properties for pouring actions," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5351–5357.

[26] Y. Kadokawa, M. Hamaya, and K. Tanaka, "Learning robotic powder weighing from simulation for laboratory automation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 2932–2939.

[27] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai *et al.*, "$\pi_{0.5}$: a vision-language-action model with open-world generalization," *arXiv preprint arXiv:2504.16054*, 2025.

[28] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair *et al.*, "Openvla: An open-source vision-language-action model," in *Proceedings of The 8th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, P. Agrawal, O. Kroemer, and W. Burgard, Eds., vol. 270. PMLR, 06–09 Nov 2025, pp. 2679–2713.

[29] X. Yuan, T. Mu, S. Tao, Y. Fang, M. Zhang, and H. Su, "Policy decorator: Model-agnostic online refinement for large policy model," in *International Conference on Learning Representations (ICLR)*, 2025.

[30] J. Luo, C. Xu, J. Wu, and S. Levine, "Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning," *Science Robotics*, vol. 10, no. 105, p. eads5033, 2025.

[31] A. Polyak, A. Zohar, A. Brown, A. Tjandra, A. Sinha, A. Lee, A. Vyas, B. Shi, C.-Y. Ma, C.-Y. Chuang *et al.*, "Movie gen: A cast of media foundation models," *arXiv preprint arXiv:2410.13720*, 2024.

[32] C. Gao, H. Zhang, Z. Xu, C. Zhehao, and L. Shao, "FLIP: Flow-centric generative planning as general-purpose manipulation world model," in *International Conference on Learning Representations (ICLR)*, 2025.