Point and Pick: Bounding-Box Conditioned Diffusion Policies and Offline RL for Target-Specific Robot Manipulation

Raul Garreta Swaroop Pal

Joshua Bowden
Stanford University
rgarreta@stanford.edu, swaroop3@stanford.edu, joshuabowden@stanford.edu

CS224R Final Project Report

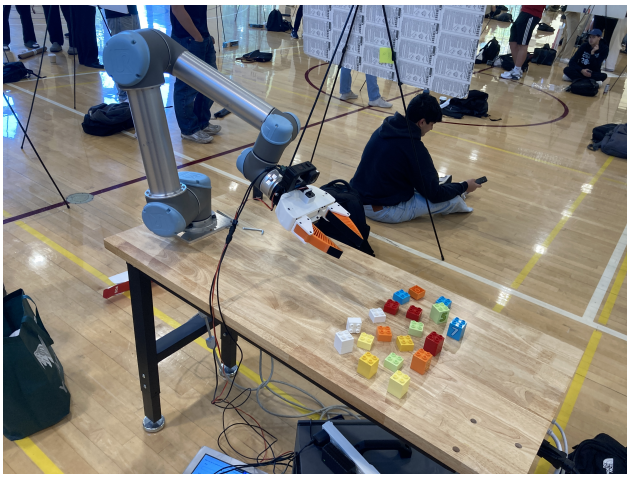


Figure 1: Our policy in action on our robot arm at the CS224R poster session.

Extended Abstract

Robotic manipulation in clutter requires more than grasping *an* object: a useful robot must grasp the *specified* object. We study target-specific LEGO picking with a UR5 arm, a wrist-mounted GoPro camera, and a UMI-style handheld data-collection pipeline. The robot observes a 224×224 wrist image, proprioception, and a 2D bounding box around the desired LEGO block, then predicts a 16-step end-effector action chunk executed through inverse kinematics. Our main hypothesis was that a lightweight task-specific diffusion policy, explicitly grounded by a visual bounding-box prompt, could outperform text-only VLA prompting for fine-grained target selection, and that offline RL could further improve robustness beyond imitation learning (IL).

Our first contribution is a real-robot IL system for point-and-pick manipulation. We collected UMI demonstrations at 60 Hz, downsampled them to 20 Hz, reconstructed tool-center-point trajectories with ORB-SLAM3, and trained diffusion policies from relative end-effector pose chunks. Early policies trained without target prompts either selected arbitrary blocks or oscillated between nearby objects. A low-dimensional bounding-box input also failed: the model largely ignored the four

coordinate values, likely because they were weak compared with high-dimensional visual features and were corrupted by crop augmentation. The key architectural change was to convert the bounding box into a binary mask channel and expand the pretrained visual patch embedding from RGB to RGB+mask, initializing RGB weights from the pretrained encoder and zero-initializing the mask-channel weights. This simple change made the prompt visually salient while preserving the pretrained image representation.

The resulting IL policy achieved strong target selection on the real robot. On a real in-distribution grid setting, the bbox-conditioned diffusion policy achieved 89% correct pick, 0% wrong pick, 11% correct contact, and 0% empty grasps. On cluttered pile scenes it achieved 76% correct pick and 24% correct contact, again with 0% wrong picks, showing that the bounding box successfully disambiguates the target. The major remaining real-world failure mode was spatial extrapolation: on far-left/far-right out-of-distribution grid positions, success fell to 19%, with 42% wrong picks and 31% empty grasps. Qualitatively, the policy often reached near the target but missed by roughly 2 cm, especially in top-down pile approaches where wrist-camera depth cues are weaker.

Our second contribution is an empirical comparison with a modern VLA baseline in simulation. To enable controlled and repeatable evaluation, we built a photorealistic MuJoCo simulator of the same task: a UR5e arm with the same UMI-style gripper picking one of twelve lego bricks on a tabletop. The simulated wrist camera is rendered at 224×224 through an equidistant fisheye warp that matches the GoPro optics, with per-episode domain randomization over lighting and material colors so the policy cannot overfit to a single look. We collected demonstrations with a scripted-IK expert and made the simulated dataset byte compatible with the real UMI dataset (identical zarr schema, channels, and image/bbox conventions), so a single training and evaluation pipeline reads both. We then ran closed-loop evaluations on held out seeds,resetting the same scene for every policy so the bbox-conditioned diffusion policy and the text-conditioned VLA are compared under identical initial conditions. We evaluated text-conditioned $\pi_{0.5}$ policies against the bbox-conditioned diffusion policy across grid, pile, random, and identical-object settings. The bbox-conditioned policy substantially outperformed the VLA in most target-specific settings, e.g. 47% vs. 3% correct pick on simulated grid ID, 40% vs. 0% on pile ID, and 47% vs. 3% on four identical objects. The VLA performed well only in the easier random-ID setting, where target descriptions were less ambiguous. These results support the central design choice: explicit visual grounding with bounding boxes is a more reliable interface than language-only prompts for selecting a specific object among nearby or identical distractors.

Our third contribution is a set of offline-RL attempts and diagnostics explaining why RL did *not* improve the policy. We trained an IQL critic on expert demonstrations and IL rollouts, then tried Q-reranking, Q-guided diffusion, structured bounded Q-guidance, structured Q-reranking, advantage-weighted diffusion fine-tuning, and a PA-RL relabeling pipeline that optimizes action chunks with a pretrained critic. These methods produced unstable, twisty, oscillatory, or upward-drifting motions rather than improved picking. A standalone critic-verification tool revealed the root cause. The critic learned an excellent state-value signal: success was ranked above miss and wrong-object outcomes with an ordering score of 1.00 over 12 checks, and V/Q increased toward successful grasps. However, it was nearly action-insensitive: sweeping actions by ± 8 cm changed Q by only 0.01–0.03 while Q varied with state by standard deviation ≈ 0.35, yielding action/state sensitivity ratios of only 0.026, 0.050, and 0.078 for x, y, and z. Expert actions coincided with the Q-argmax in less than 1% of successful pre-grasp frames. Thus, the critic learned "which states are good" but not "which actions move the robot to good states," because the offline dataset had narrow action support. Our main lesson is that critic-based reranking or guidance cannot help unless the critic is trained with sufficient action diversity, counterfactual perturbations, denser/Monte-Carlo targets, or a tighter trust region.

Overall, the project produced a working real-robot bbox-conditioned manipulation policy, a

matched simulation/VLA evaluation, and a negative but actionable RL result. The strongest result is that a small, visually grounded diffusion policy can provide a practical point-and-pick manipulation interface. The central limitation is that offline RL over narrow demonstration and rollout data produced value functions useful for diagnosis but not for action optimization. Future work should combine the bbox-conditioned policy with action-diverse critic training, residual correction policies constrained to the IL manifold, broader data at workspace extremes, and improved depth cues for top-down grasps.

Abstract

We study target-specific robotic manipulation: given a wrist-camera image and a 2D bounding box around a desired LEGO block, a UR5 robot must pick that block from clutter. We build on UMI-style real-world data collection and diffusion-policy imitation learning, adding a visual bounding-box mask channel to ground the policy in a selected target. The resulting real-robot policy achieves 89% correct picks on an in-distribution grid and 76% correct picks in cluttered pile scenes, with 0% wrong-object picks in both settings. In simulation, the bbox-conditioned diffusion policy outperforms a text-conditioned $\pi_{0.5}$ VLA baseline on most target-specific settings, especially when distractors are nearby or identical. We then investigate whether offline RL can improve the IL policy. Despite implementing IQL-based Q-reranking, Q-guided diffusion, structured bounded guidance, advantage-weighted fine-tuning, and PA-RL action relabeling, RL did not improve real-world behavior and often produced unstable or drifting motion. Diagnostics show why: the IQL critic ranks states by outcome very well but is almost flat with respect to action perturbations, making it unsuitable for reranking or gradient guidance. Our results suggest that explicit visual grounding is highly effective for target selection, while offline RL for diffusion manipulation policies requires action-diverse data or counterfactual critic training before critic-based action optimization can be useful.

1 Introduction

Many tabletop manipulation tasks require a robot to pick a specific object among several nearby distractors. A natural high-level interface is to identify the target visually—for example, by clicking on it or by using a detector/VLM to produce a bounding box—and then invoke a low-level manipulation policy conditioned on that target. This setting is distinct from generic grasping: a policy that reliably grasps any visible object can still fail the task if it picks the wrong object.

We study this problem with a real UR5 robot, a wrist-mounted GoPro camera, and a UMI-style handheld demonstration device. The task is to pick a designated LEGO block from either a grid or a cluttered pile. The policy receives the wrist image, end-effector proprioception, gripper width, and a bounding box identifying the target. It outputs a short chunk of end-effector deltas that is executed using inverse kinematics.

Our project has two goals. First, we aim to build a practical point-and-click manipulation policy that can select a specific object in clutter using a lightweight diffusion architecture rather than a large VLA. Second, we ask whether offline RL can improve the policy beyond imitation learning, especially on failures caused by missed grasps and out-of-distribution target positions.

The main findings are:

- Bounding-box conditioning works best when injected as a visual mask channel. A four-coordinate low-dimensional bbox input was ignored, while an RGB+mask visual encoder enabled robust target selection.
- The bbox-conditioned diffusion policy achieved strong real-robot results: 89% correct pick on grid ID and 76% correct pick on pile ID, with 0% wrong-object picks in both settings.
- A text-conditioned $\pi_{0.5}$ VLA baseline performed poorly on most target-specific simulated settings, suggesting that explicit visual grounding is more reliable than text-only prompting for nearby or identical distractors.
- Offline RL did not improve the policy. The IQL critic learned useful state values but was too action-insensitive to guide diffusion samples, rerank candidate chunks, or relabel actions safely.

2 Related Work

Diffusion policies for manipulation. Diffusion Policy models robot action sequences through iterative denoising has shown strong performance on multimodal visuomotor imitation-learning tasks [3]. This makes it a natural base policy for our setting, where multiple grasp approaches can be valid and where action chunks smooth closed-loop control. Our work uses a diffusion policy as the low-level controller, but modifies the observation representation so the policy can condition on an explicit object target.

UMI-style real-world data collection. The Universal Manipulation Interface (UMI) enables efficient real-world data collection by recording handheld demonstrations and reconstructing trajectories for robot execution [4]. We follow this setup closely: demonstrations are collected with a UMI-style handheld device using the same camera as deployment, trajectories are reconstructed with SLAM, and the policy predicts relative TCP deltas. Our contribution is to extend this pipeline to target-specific picking with bbox prompts and to evaluate both IL and offline-RL variants.

Visual prompting and bounding-box conditioned policies. Recent work has explored visual prompts and bounding boxes as interfaces for robotic manipulation [11, 8]. These approaches motivate our core design choice: instead of asking the policy to infer the target from a language string, we provide an explicit spatial prompt. Our experiments show that the representation of this prompt matters: the bbox must be made visually salient to the image encoder.

Vision-language-action models. VLA models such as RT-1 and $\pi_{0.5}$ condition action generation on natural-language instructions and large-scale visual priors [2, 10]. They offer broad generalization due to being trained on internet-scale text video, and robot actions [5], but fine-grained spatial grounding through text instructions can be unreliable when several similar objects are nearby. We compare against a text-conditioned $\pi_{0.5}$ baseline and find that, for our target-specific LEGO task, bbox conditioning is substantially more reliable than language-only prompting in most simulated settings.

Offline RL for imitation-learned policies. Offline RL methods such as IQL learn value functions from fixed datasets without online exploration [6]. Several robotics approaches combine behavior cloning with value-weighted losses or policy improvement under a learned critic [9]. We test this family of ideas in the context of diffusion action chunks: Q-reranking, Q-guided denoising, advantage-weighted BC, and PA-RL-style [7] action relabeling. Our negative result highlights a key support issue: if the dataset contains only a narrow action distribution, a critic can learn state values without learning action sensitivity.

3 Task and System Setup

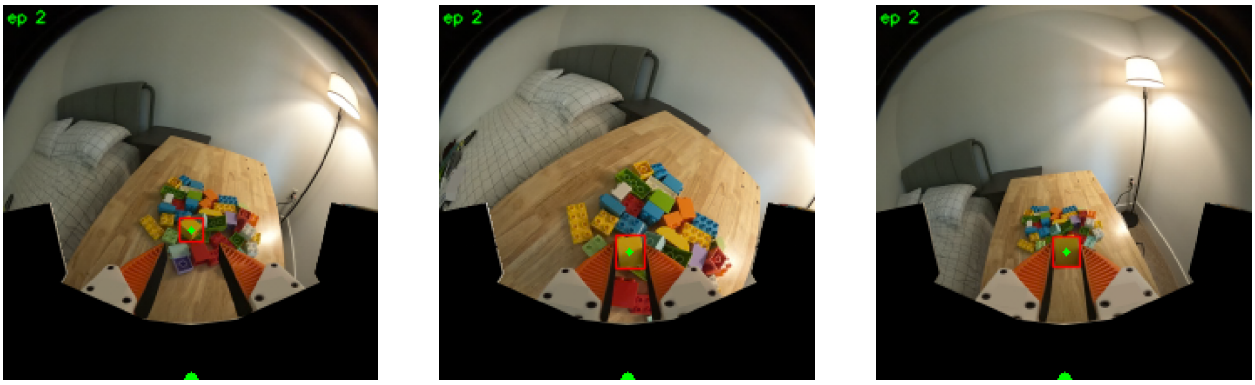
3.1 Task Definition

The task shown in Figure 2 is to pick a target LEGO block from a tabletop scene. The target is specified by a bounding box in the wrist-camera image. The robot begins from a medium-height pose above the workspace with the gripper open, approaches the target, descends while aligning, closes the gripper, and returns to the starting pose while lifting the object.

At each control step, the policy observes:

- a wrist-camera RGB image resized to 224×224;
- a bounding box $[x_0, y_0, x_1, y_1]$ around the target;
- end-effector position and orientation in task space;
- gripper opening width.

The policy predicts a 16-step action chunk of relative end-effector pose and gripper-width commands. Commands are executed on the UR5 through inverse kinematics.



Approach target block prompted by the bbox.

Pick target block prompted by the bbox.

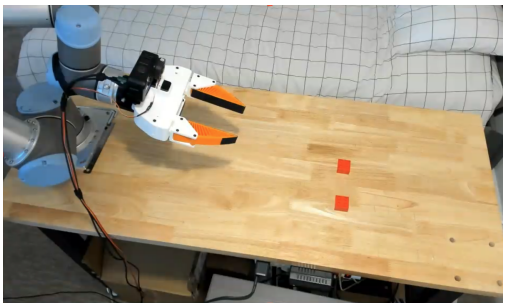
Return to start position with picked block.

Figure 2: Task definition with example camera views from real data collected with UMI.

3.2 Real and Simulated Environments

Figure 3 summarizes the real system. Demonstrations are collected using a 3D-printed UMI-style handheld device equipped with the same GoPro camera used during deployment. Videos are recorded at 60 Hz and downsampled to 20 Hz for training. We use ORB-SLAM3 to estimate the camera trajectory and convert it into TCP poses relative to an ArUco-defined workspace origin. During training, trajectories are transformed into deltas relative to the first frame of each action chunk.

Figure 4 shows the simulated environment that mirrors the real-world setup: a UR5 with a UMI-style parallel-jaw gripper must pick a specified Lego brick from a 3×4 tabletop grid using only a wrist-mounted GoPro-like RGB view and a per-frame 2D bounding box prompt, with no segmentation or 3D state available at inference. The simulator is implemented with MuJoCo and uses Menagerie UR5e and gripper assets, textured room geometry, realistic lighting, and 12 colored DUPLO meshes, while the rendered wrist camera approximates GoPro Hero 10 optics through wide-FOV rendering, supersampling, downsampling, and fisheye warping. The simulated data collector is built so its recorded dataset is compatible with the real UMI dataset (identical schema, channels, gripper width, and a bounding box projected through the same fisheye warp), and per-episode domain randomization over lighting and material colors reduces the sim-to-real gap, so we could potentially use a single pipeline can train and evaluate on both sources and can have a robust policy.



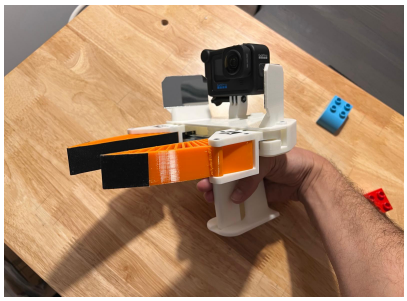
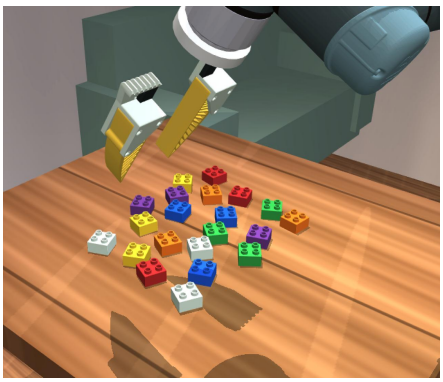


Figure 3: Left: Robot setup: UR5 arm + actuated gripper + GoPro Hero10 camera + Elgato HD 60X video capture card. Right: UMI handheld data collection device.



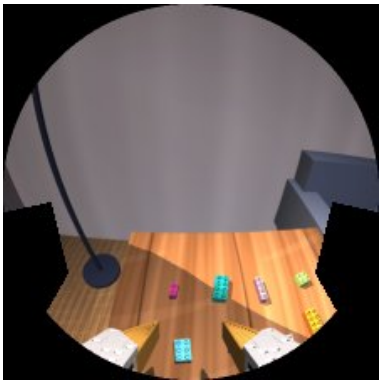


Figure 4: Left: Simulated robot setup. Right: Simulated view from wrist.

4 Method

4.1 BBox-Conditioned Diffusion Policy

Our base policy is a diffusion action-chunk model conditioned on visual and proprioceptive observations. During training, Gaussian noise is added to the ground-truth action chunk and the network predicts the denoising target. During inference, the model iteratively denoises a sampled action chunk and executes the resulting relative TCP commands.

The central architectural change is bbox-as-mask conditioning. We convert the bounding box into a binary mask image aligned with the RGB observation, concatenate it as a fourth image channel, and expand the visual patch embedding from RGB to RGB+mask. The pretrained RGB weights are copied into the first three channels; the new mask-channel weights are initialized to zero. This makes the modification initially non-disruptive while allowing the model to learn target-specific visual attention.

The following are the hyperparameters and configurations used to train the diffusion policy:

- **Visual encoder:** ViT-Large encoder with wrist-camera RGB observations.
- **Video preprocessing:** demonstrations were recorded at 60 Hz and downsampled to 20 Hz for policy training.
- **Action horizon:** the policy predicts 16-step action chunks.
- **Inference:** 16 DDIM inference steps were used at deployment, matching the action chunk size used in UMI.

- **Image augmentations:** random crop and color jitter.
- **Optimizer:** AdamW with learning rate 3.0×10^{-4} and weight decay 1.0×10^{-6}.
- **Compute:** training used 4× NVIDIA A100 40GB GPUs.
- **Batch size:** 128 samples per GPU, for an effective batch size of 512.
- **Object Tracking:**

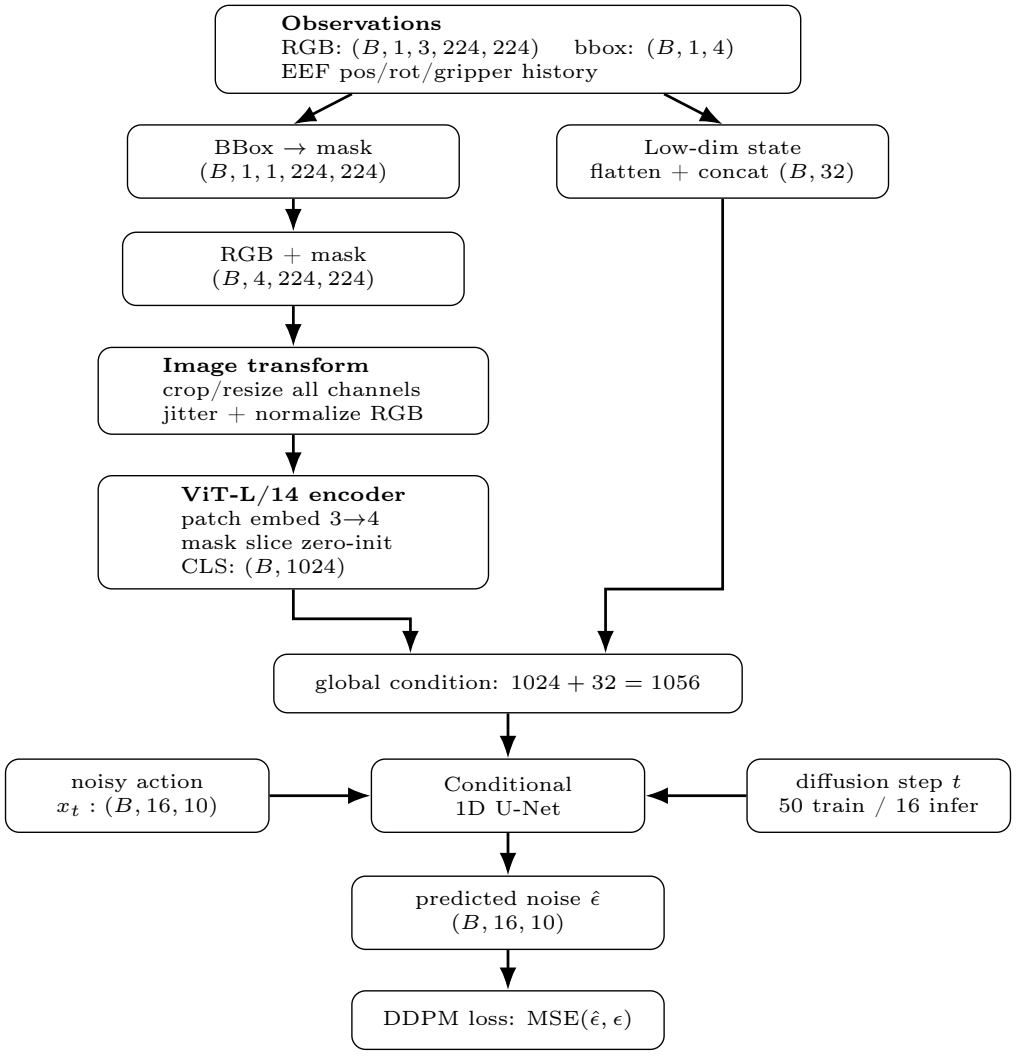


Figure 5: BBox-conditioned diffusion policy. The key change is converting the bounding box into a mask channel before the visual encoder.

4.2 Offline RL Layer

After training the IL policy, we trained an IQL critic on a dataset containing expert demonstrations and IL-policy rollouts, including successes, misses, wrong-object attempts, and correction trajectories. Rewards were sparse and outcome-based: successful target grasp-and-lift episodes received positive terminal reward, while misses and wrong-object outcomes were labeled as failures. The IQL objective

learns a value function $V(s)$ and an action-value function $Q(s, a)$, and we use the advantage $A(s, a) = Q(s, a) - V(s)$ for policy improvement attempts.

We implemented four families of critic-based policy improvement:

1. **Q-reranking:** sample multiple diffusion action chunks, score each with the IQL critic, and execute the chunk with maximum predicted Q.
2. **Q-guided diffusion:** use gradients of $Q(s, a)$ during denoising to bias samples toward high-value action chunks.
3. **Advantage-weighted diffusion fine-tuning:** weight the diffusion denoising loss by $\exp(A(s, a)/\lambda)$, imitating high-advantage chunks more strongly while remaining close to behavior cloning.
4. **PA-RL action relabeling:** optimize candidate action chunks under the critic, cache optimized actions in the dataset, and fine-tune the diffusion policy against these relabeled targets.

Because early variants produced unstable motion, we added increasingly conservative stabilizers: bounded per-dimension residuals, cosine ramps over the final part of the chunk, structured candidate grids over $x/y/z$/rotation/gripper dimensions, sign-flip diagnostics for local-vs-world frame mismatches, BC anchoring to a frozen reference policy, and revert-on-no-improvement guards for local Q-gradient ascent.

4.3 Simulation

To get repeatable evaluation and cheap data, we built a MuJoCo version of the task with the same UR5e arm and UMI-style gripper picking one of twelve colored bricks. Demonstrations are generated by a scripted inverse-kinematics expert (a `mink` QP-IK solver) that reaches, descends, grasps, and lifts the target, and we keep only successful episodes. The wrist camera is rendered to approximate the GoPro Hero 10 optics through wide field-of-view rendering, supersampling, downsampling to 224×224, and a fisheye warp, and per-episode domain randomization varies lighting, tint, and material colors so the policy cannot overfit to a single look. The simulated dataset is recorded in the same format as the real UMI data, with the same table-anchored TCP frame, metric gripper width, and bounding box projected through the same fisheye warp, so a single pipeline trains and evaluates on both.

We ran these experiments on Modal, which hosted data collection, training, inference, and evaluation as cloud jobs: demonstrations were collected with a CPU fan-out of parallel workers, and policies were trained on GPU jobs with the same recipe as the real policy. At evaluation time each policy is exposed as a Modal inference endpoint, the bbox-conditioned diffusion policy from its checkpoint and the $\pi_{0.5}$ VLA from a separate fine-tuned endpoint, so both answer over the same interface. A closed-loop driver resets the same MuJoCo scene with held out seeds, queries the chosen policy, and executes the returned action chunks through the `mink` QP-IK solver, keeping evaluation identical across policies. We also built a FastAPI dashboard backed by a single MuJoCo worker thread that lets a user load a scene, draw or select the target bounding box, switch between the diffusion policy and the VLA, and watch the rollout, which we used for debugging and for side-by-side comparison on the same scene.

5 Experiments

5.1 Real-World IL Development

The real-world IL experiments followed a curriculum from simple single-block picking to cluttered target-specific picking. Table 1 summarizes the main stages. The most important transition was

Exp. 6 to Exp. 7: bbox coordinates as low-dimensional state were not enough, but bbox-as-mask conditioning worked once the dataset required target disambiguation. All episodes are collected using the UMI handheld device.

Exp.	Data / change	Result / lesson
1	Early pile/single-block data without bbox.	Policy picked arbitrary blocks or oscillated between nearby objects.
2	10 demonstrations on simplified single-object task.	Too little data for reliable behavior.
3a	75 single-block demonstrations with left-right variation; ViT encoder.	Improved, but sensitive to start height and gripper opening.
3b	ResNet-34 visual encoder ablation on the same data.	Worse validation MSE than ViT; not used for final policy.
4	175 single-block grid demonstrations.	Better spatial coverage and cross-color generalization, but multiple-object scenes remained OOD.
5	486 single-block demonstrations with recovery/correction cases and mirror augmentation.	Strong non-bbox baseline, but still confused by clutter.
6	Same 486 episodes with automatically tracked bbox as four low-dimensional inputs.	Mechanically worked, but the policy mostly ignored the bbox.
7	Two same-color blocks per scene; bbox converted to visual mask channel.	Bbox began to direct the policy toward the selected target.
8	Cluttered pile scenes with bbox mask identifying the picked block. 184 real episodes, mirrored vertically, 368 total	Strong target-specific performance; remaining misses were mostly depth/reach errors.

Table 1: Real-world IL curriculum and key lessons.

5.2 Simulated IL Development

After validating the model architecture and training recipe with the real environment, we collected 300 simulated episodes using a heuristic policy and train a diffusion model with the same recipe as the real environment. We evaluated both policies across six scene settings that probe target selection under increasing ambiguity. *Grid ID* places bricks on the same 3×4 grid seen in training; *Grid OOD* pushes brick positions toward the far edges of the workspace; *Pile ID* uses a cluttered pile; *Random ID* scatters bricks at random table positions; and *Identical 2* and *Identical 4* place two or four bricks of the same color, so only the bounding box, not color or coarse location, can identify the target. Each setting was run for 30 closed-loop episodes on held-out seeds (`seed` ≥ 10000), and every policy was evaluated on the same reset scenes so the comparison is matched episode for episode.

For each episode the diffusion policy received the target's bounding box, while the VLA received a text prompt naming the target's color and coarse location (for example "pick up the red block on the left"). Both policies ran closed-loop through the same `mink` QP-IK controller, and we scored each rollout with the four mutually exclusive outcomes defined in Section **??** (Pick, Wrong, Contact, Empty). We ran the rollouts with a parallel evaluation harness so all settings and both policies could be swept under identical conditions.

5.3 IQL Critic Training

We trained an *Implicit Q-Learning* (IQL) critic offline on RL-annotated UMI demonstration data, using a dataset of 858 episodes composed by a mixture of the expert data collected with the UMI (368 episodes used to train the IL policy) and 490 episodes obtained from rollouts of the IL policy which includes successful and unsuccessful outcomes. Figure 6 shows statistics of the dataset.

Rather than learning a policy directly, the run fits a value function $V(s)$ and twin action-value functions $Q_1(s, a), Q_2(s, a)$ over fixed action chunks. The IQL objective uses an asymmetric *expectile* regression on $Q - V$ with expectile $\tau = 0.7$, which pushes V toward the upper expectile of Q and

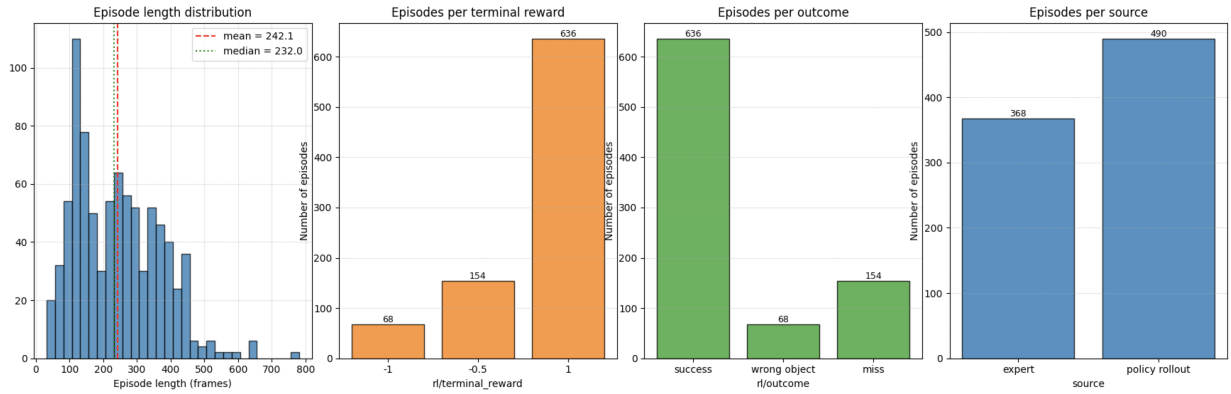


Figure 6: Statistics for IQL dataset.

thus implicitly selects high-value actions without querying out-of-distribution actions. Target Q networks are maintained by Polyak averaging (rate 0.005), with discount $\gamma = 0.99$.

The critic reuses the diffusion policy's visual encoder backbone so the state encoding is byte-identical to the BC policy. The visual encoder is **frozen** and **warm-started** from the pretrained diffusion-policy checkpoint, ensuring the critic operates in the same visual feature space as the behavior policy. The state is projected to a 512-dim embedding; the flattened (H=16, A=10) action chunk is projected to a 256-dim embedding through a small MLP. The V and Q heads are two-layer 512-wide MLPs.

AdamW is used with learning rate 4×10^{-4} (encoder LR scaled by 0.1), weight decay 10^{-4}, cosine schedule with 150 warmup steps, for 30 epochs. Validation runs on a 10% validation split held out every epoch and bucketed critic diagnostics run every 3 epochs, logging $V/Q/A$/TD-error grouped by episode outcome (success / miss / wrong-object) and phase (start, mid, near terminal, terminal), along with ordering-correctness scores.

5.4 Evaluation Metrics

We report four mutually exclusive rollout outcomes:

- **Pick:** the robot grasps and lifts the correct target block.
- **Wrong:** the robot grasps and lifts a non-target block.
- **Contact:** the robot contacts or partially grasps the correct target but does not lift it successfully.
- **Empty:** the robot closes on empty space or fails to make meaningful contact.

5.5 Real-Robot IL Results

Table 2 shows the real-robot evaluation of the bbox-conditioned diffusion policy. The policy is highly reliable in-distribution and in cluttered pile scenes, with no wrong-object picks in either setting. Its main weakness is out-of-distribution spatial extrapolation: far-left and far-right target positions produce misses and wrong-object picks. Remaining failures are interpretable: mostly depth estimation, collisions of the gripper's fingers with other blocks. It is important to note that in distribution, the policy achieves 100% correct target contact, showing that the instruction following is very robust.

Setting	Policy	Pick	Wrong	Contact	Empty
Grid ID	IL Diffusion BBox	89%	0%	11%	0%
Grid OOD	IL Diffusion BBox	19%	42%	8%	31%
Pile ID	IL Diffusion BBox	76%	0%	24%	0%

Table 2: Real-world policy evaluation. BBox-conditioned diffusion is strong in distribution but weak at far workspace extremes.

5.6 Simulated IL and VLA Baseline Results

We also built a MuJoCo simulation environment aligned with the real robot setup, including a UR5 arm, wrist-camera view, gripper geometry, and LEGO-like objects. The simulator enabled a controlled comparison against a $\pi_{0.5}$ VLA baseline using text prompts such as "pick up the red block on the left." Table 3 summarizes the simulated evaluation from the poster.

Setting	Policy	Pick	Wrong	Contact	Empty
Grid ID	IL Diffusion BBox	47%	0%	30%	23%
	VLA (color+loc)	3%	3%	10%	83%
Grid OOD	IL Diffusion BBox	17%	0%	27%	57%
	VLA (color+loc)	3%	10%	0%	87%
Pile ID	IL Diffusion BBox	40%	0%	37%	23%
	VLA (color+loc)	0%	7%	0%	93%
Random ID	IL Diffusion BBox	60%	0%	13%	27%
	VLA (color+loc)	83%	0%	0%	17%
Identical 2	IL Diffusion BBox	47%	0%	20%	33%
	VLA (color+loc)	37%	43%	3%	17%
Identical 4	IL Diffusion BBox	47%	0%	17%	37%
	VLA (color+loc)	3%	23%	0%	73%

Table 3: Simulation comparison between bbox-conditioned diffusion and text-conditioned VLA prompting. The VLA is competitive only in the random-ID setting and struggles with fine-grained target selection in grid, pile, and identical-object settings.

These results suggest that explicit visual grounding is especially useful when the target cannot be described unambiguously by color or coarse location. In identical-object settings, the bounding box specifies the instance directly, while language prompts must rely on spatial and color descriptors that can be hard to interpret in complicated environments.

6 Offline RL Results and Diagnostics

6.1 Q-Reranking and Q-Guided Diffusion

The first RL attempts used the IQL critic at inference time. Q-reranking generated multiple diffusion action chunks and selected the candidate with the highest critic value. Q-guided diffusion added gradients from the critic during the denoising process. Neither method improved over pure IL. Candidate samples were often too similar to change the outcome, while higher diffusion temperature and Q-gradient guidance produced twisty, unstable trajectories.

We then implemented structured variants to make the search space more interpretable. Instead of relying only on stochastic diffusion samples, we perturbed the base IL chunk along individual dimensions and Cartesian products of per-axis offsets. We also added per-axis sign-flip diagnostics after observing that end-effector-local motion could invert world-frame directions due to the TCP yaw. Despite these changes, the selected correction often flip-flopped from cycle to cycle, producing oscillation rather than stable reaching to far-left or far-right targets.

6.2 Advantage-Weighted and BC-Anchored Fine-Tuning

We next tried advantage-weighted fine-tuning, where high-advantage action chunks receive larger denoising-loss weights. To stabilize training, we added an unweighted BC term and an MSE anchor to a frozen BC reference policy. This was intended to preserve the IL manifold while still biasing the policy toward high-value actions. In real-robot rollouts, however, the policy continued to produce unstable or twisted movements. The anchor reduced but did not eliminate drift.

6.3 PA-RL Action Relabeling

Finally, we implemented a policy-agnostic RL (PA-RL) [7] pipeline that optimizes dataset actions under the critic and fine-tunes the diffusion policy on the optimized action cache. The implementation included:

- a shared IQL critic-loading utility;
- a PA-RL action optimizer with candidate proposal, global Q-reranking, local Q-gradient ascent, and revert-on-no-improvement;
- dataset support for cached optimized actions and validity masks;
- diffusion-loss modifications to train on optimized actions when present;
- periodic relabeling in the training workspace;
- performance optimizations including bf16 autocast, observation-encoder deduplication across candidates, GPU-resident caches, and vectorized scatter writes.

Training diagnostics showed the expected PA-RL signature: mean chosen Q increased, global Q spread collapsed, local optimization was often reverted, and the dataset action was rarely selected. However, real-world evaluation showed systematic upward end-effector drift. Validation loss decreased because it was measured against optimized targets, but dataset-action drift metrics increased after roughly epoch 6. Even with an anchor coefficient of 0.5, the policy retained most of the optimized-target bias and continued drifting upward on the robot.

6.4 Critic Verification: The Root Cause

To understand these failures, we built a standalone critic-verification tool with two tests.

Test A: value ordering and temporal ramp. The critic passed this test. It ranked success above miss and wrong-object outcomes with ordering score 1.00 over 12 checks. The success-vs-miss and success-vs-wrong margins were approximately +0.62 and +0.88 overall, and roughly +1.0 near terminal states. Its V and Q values ramped upward toward successful grasps.

Test B: action sensitivity. The critic failed this test. Sweeping actions by ±8 cm in x/y/z changed Q by only 0.01–0.03, while Q varied by standard deviation ≈ 0.35 across states. The action/state sensitivity ratios were 0.026 for dx, 0.050 for dy, and 0.078 for dz. On successful pre-grasp frames, the expert action matched the Q-argmax less than 1% of the time. Table 4 summarizes the diagnosis.

Diagnostic	dx	dy	dz
Q change under ±8 cm sweep	0.01–0.03	0.01–0.03	0.01–0.03
Across-state Q std.		≈ 0.35	
Action/state sensitivity ratio	0.026	0.050	0.078
Verdict	Too flat	Too flat	Too flat

Table 4: IQL critic action-sensitivity diagnosis. The critic is useful as a state-value estimator but too flat with respect to actions for reranking, guidance, or relabeling.

This explains the RL failures. The critic learned which states look like future success, but it did not learn how action choices affect transitions into those states. Because the offline dataset contained expert and policy actions in a narrow support region, the critic had little evidence to assign meaningful Q differences to counterfactual action perturbations. As a result, reranking and gradient ascent optimized against critic artifacts rather than real action quality.

7 Discussion

Why bbox-as-mask worked. The failure of low-dimensional bbox conditioning and the success of mask conditioning suggest that the bottleneck was not the lack of target information, but the way the target information entered the network. A four-dimensional coordinate vector can be overwhelmed by high-dimensional visual features. A mask channel instead makes the target spatially aligned with the image and lets the visual encoder learn local correspondences between the target region, the gripper, and the action.

Why the policy still fails OOD. The real-grid OOD result indicates that the policy has not learned full workspace generalization. It likely interpolates well within demonstrated spatial support but extrapolates poorly near the extremes. This is consistent with the IL nature of the method: the policy is optimized to reproduce demonstrated action chunks and has limited incentive to recover from states or target locations not well covered by the data.

Why offline RL failed. The RL results are negative but informative. The IQL critic has strong state-value signal, which is useful for diagnosing whether a trajectory is likely to succeed. However, action optimization requires local action sensitivity. Without action-diverse data, counterfactual negatives, or a dynamics-aware target, the critic is underconstrained off the behavior manifold. This makes Q-guided diffusion and PA-RL relabeling unsafe: they amplify critic biases in action dimensions the critic does not truly understand.

Implications for future RL. The next RL attempt should not be more tuning of reranking or guidance. It should first fix critic training. Promising directions include adding action perturbations with learned or heuristic labels, using Monte-Carlo returns rather than bootstrapped sparse TD

targets for short episodes, training residual policies constrained by hard trust regions, adding per-dimension action anchors, collecting targeted data at workspace extremes, and using VLMs for denser rewards [1]. For the upward-drift failure, a specific diagnostic is to evaluate $Q(s, a_{\text{dataset}})$ vs. $Q(s, a_{\text{dataset}} + \Delta\hat{z})$ and $\partial Q/\partial a_z$ at descent states.

8 Limitations

The project has several limitations. First, real-world evaluation was limited by the time cost of physical rollouts, so confidence intervals are not yet included. Second, the real and simulated evaluations are not perfectly matched; the VLA comparison is currently strongest as a controlled simulation baseline rather than a full real-world benchmark. Third, the reward labels are sparse, which makes critic learning difficult from narrow offline data. Fourth, the wrist-only camera setup makes depth estimation difficult for top-down grasps over a cluttered pile. Finally, we only focus on simple block shapes and have not explored more complex shapes that are harder to segment or grasp.

9 Conclusion

We built and evaluated a target-specific manipulation system that uses a bounding box as a visual prompt for a diffusion policy. The final IL policy demonstrates that a lightweight RGB+mask diffusion policy can produce a practical point-and-click manipulation interface on a real UR5 robot, achieving high in-distribution success and avoiding wrong-object picks in key real-world settings. Compared with a text-conditioned VLA baseline in simulation, explicit bbox grounding is substantially more reliable for fine-grained target selection among nearby or identical objects.

We also found that offline RL did not improve the policy despite several increasingly constrained implementations. The main reason is that the IQL critic learned state value but not action sensitivity. This result is useful for future work: critic-guided diffusion policy improvement should be attempted only after broadening action support or training the critic with counterfactual action diversity. In short, bbox-conditioned IL solved the target-selection part of the problem; improving recovery and OOD reaching will require better data coverage and action-sensitive value learning.

Team Contributions

The final division of labor changed from the proposal because the real-robot IL system reached a working target-conditioned policy earlier than expected, while the RL layer and simulation/VLA comparison required substantially more debugging and diagnosis than initially planned.

Raul Garreta. Raul led the real-robot system and RL integration. He collected real UMI demonstrations, implemented the bbox annotation/tracking pipeline, implemented and trained the bbox-conditioned diffusion policy, ran real-robot evaluations, implemented IQL-based Q-reranking and Q-guided diffusion experiments, developed structured guidance/reranking variants, implemented BC-anchored AWR fine-tuning and the PA-RL relabeling pipeline, and built the critic-verification diagnostics that identified the action-insensitivity failure mode. He ran the jobs to train the diffusion policy both with the real and simulated datasets and ran the jobs to learn the Q and V functions with IQL plus fine tuning the policy with it. He helped define the matched evaluation protocol for comparing text-conditioned VLA policies against bbox-conditioned diffusion policies.

Swaroop Pal. Swaroop led the simulation environment and sim-policy experiments. He built the MuJoCo UR5e setup with the UMI-style gripper and a wrist camera that models the GoPro Hero 10 optics through wide field-of-view rendering, supersampling, and a fisheye warp. He wrote a scripted inverse-kinematics demonstrator (`mink` QP-IK) to generate training data, aligned the simulated observations and actions with the real UMI schema, and added domain randomization for sim/real visual alignment. He trained ACT and diffusion policies with and without bbox conditioning, set up the experiments on Modal(data collection, training, inference, and policy serving), and built the closed-loop evaluation that resets the same held-out scenes for every policy. He ran the six evaluation settings and the matched comparison of the bbox-conditioned diffusion policy against the text-prompted $\pi_{0.5}$ VLA, and implemented the dashboard for switching policies and providing bbox inputs.

Joshua Bowden. Joshua led the VLA baseline. He set up the $\pi_{0.5}$ fine-tuning environment, adapted UMI data to the VLA input/output interface, handled wrist-camera slot mapping and state/action padding, and fine-tuned $\pi_{0.5}$ under multiple initialization strategies (base vs DROID pretraining). He trained VLA models on the real-world UMI dataset with generic "pick up the lego" descriptions using Modal. He used inverse kinematics to convert UMI tool space actions to simulation joint space action, as well as developed Mujoco and gripper-specific transforms to conduct evaluations in the grid simulation environment using Modal. He also labeled the UMI dataset with detailed spatial-color descriptions to finetune with specific language descriptions.

AI Disclosure

We used Claude to help write scripting to configure the UR arm in the Mujoco simulation environment, to help adapt our code to run on Modal cloud, and to brainstorm for ways to make our simulation baselines closer to real life evaluation.

References

[1] Kate Baumli, Satinder Baveja, Feryal Behbahani, Harris Chan, Gheorghe Comanici, Sebastian Flennerhag, Maxime Gazeau, Kristian Holsheimer, Dan Horgan, Michael Laskin, Clare Lyle, Hussain Masoom, Kay McKinney, Volodymyr Mnih, Alexander Neitz, Dmitry Nikulin, Fabio Pardo, Jack Parker-Holder, John Quan, Tim Rocktäschel, Himanshu Sahni, Tom Schaul, Yannick Schroecker, Stephen Spencer, Richie Steigerwald, Luyu Wang, and Lei Zhang. Vision-language models as a source of rewards, 2024.

[2] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems (RSS)*, 2023.

[3] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.

[4] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Robotics: Science and Systems (RSS)*, 2024.

[5] Alexander Khazatsky et al. Droid: A large-scale in-the-wild robot manipulation dataset, 2025.

[6] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

[7] Max Sobol Mark, Tian Gao, Georgia Gabriela Sampaio, Mohan Kumar Srirama, Archit Sharma, Chelsea Finn, and Aviral Kumar. Policy agnostic rl: Offline rl and online rl fine-tuning of any class and backbone, 2024.

[8] Muhammad A. Muttaqien, Tomohiro Motoda, Ryo Hanai, and Yukiyasu Domae. Visual prompting for robotic manipulation with annotation-guided pick-and-place using act. *arXiv preprint arXiv:2508.08748*, 2025.

[9] Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

[10] Physical Intelligence, Kevin Black, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[11] Yihao Wu, Jinming Ma, Junbo Tan, Yanzhao Yu, Shoujie Li, Mingliang Zhou, Diyun Xiang, and Xueqian Wang. Learning to manipulate anything: Revealing data scaling laws in bounding-box guided policies. *arXiv preprint arXiv:2602.11885*, 2026.