# Beyond Success: Refining Elegant Robot Manipulation from Mixed-Quality Data via Just-in-Time Intervention

Supplementary Material

## A. Overview

Considering the space limitation of the main paper, we provide additional details and results in this supplementary appendix. Specifically, we present comprehensive descriptions of the **LIBERO-Elegant Benchmark**, detailed **implementation settings** for all baselines, and extended **experiment configurations** in both simulation and real-world environments. This appendix is organized as follows:



## B. LIBERO-Elegant Benchmark Details

### B.1. Motivation and Construction

**Motivation.** Existing LIBERO [17] datasets cover a wide range of manipulation tasks and provide rich demonstration data. However, they evaluate performance only by *whether* the goal is achieved. In practice, many demonstrations succeed but differ noticeably in how well they satisfy implicit task expectations. Such variation is currently ignored, making it difficult to measure and improve execution quality [13, 24].

**Analysis of Existing LIBERO Suites.** The LIBERO benchmark consists of four task groups, each targeting a distinct generalization dimension:

- **LIBERO-Spatial:** Tests spatial generalization, performing the same task under different spatial layouts.
- **LIBERO-Object:** Tests object generalization, executing the same goal with different target objects.
- **LIBERO-Goal:** Tests goal generalization, covering diverse goal types.
- **LIBERO-100:** Combines all the above variations to evaluate compositional generalization.

These suites are well suited for evaluating task completion across diverse conditions, but they still treat all successful executions as equivalent.

**Motivated Subset Construction.** To fill this gap, we introduce **LIBERO-Elegant**, a curated subset where the *quality of execution* becomes an explicit evaluation dimension. Tasks in LIBERO-Elegant are chosen because they rely on Implicit Task Constraints (ITCs) that influence behavioral quality, enabling more fine-grained comparison among successful trajectories. This provides a controlled environment for studying methods that refine policy behavior without changing the underlying goal.

### B.2. Task List and Structure

**Overview.** The LIBERO-Elegant benchmark consists of eight manipulation tasks selected for their sensitivity to motion quality and implicit task constraints (ITCs). Each task inherits the base instruction and success condition from the original LIBERO suite but introduces additional qualitative criteria that emphasize how the task is performed rather than merely whether it succeeds.

**Task Composition.** To cover a diverse range of execution qualities, the eight tasks are grouped according to four core *Elegance Criteria*:

- **Task Sequence Integrity.** Whether the execution respects the intended ordering and timing of key actions without premature releases, unnecessary pauses, or unintended reversals. This dimension ensures that objects remain securely grasped until the correct release moment (Tasks 0–1) and that the overall action sequence progresses smoothly and purposefully toward the goal.
- **Target Pose Accuracy.** Whether the manipulated object reaches the correct final position within a tight spatial tolerance. This dimension captures precise placement requirements such as centering a bowl on a plate or positioning a frying pan accurately on a stovetop (Tasks 2–3).
- **Pose Alignment.** Whether the object's final orientation matches the desired orientation for stable insertion or placement. This dimension evaluates rotational correctness, as required when inserting a book with proper orientation into a caddy compartment (Tasks 4–5).
- **Collision-Free Execution.** Whether the trajectory avoids unintended contact with the environment or nearby objects. This dimension assesses the safety and spatial awareness of the motion, ensuring smooth push-

ing without collisions (Task 6) and transporting objects through cluttered spaces without touching neighboring items (Task 7).

**Summary.** This categorization ensures broad manipulation coverage, requiring policies to perform not only successful but also precise, stable, and safe executions across diverse physical constraints.

**Implicit Task Constraints as Concrete Elegance Rules.** While the Elegance Criteria describe high-level dimensions of execution quality, the actual evaluation in LIBERO-Elegant is grounded in task-specific *Implicit Task Constraints* (ITCs), which define *how* elegance is judged for each task. Each task is associated with a primary Elegance Criterion (last column of Table 4), and its corresponding ITC (third column) specifies a concrete, verifiable rule that reflects this criterion. For example, "no premature release before the object is fully inside the container" instantiates *Task Sequence Integrity*, while "centering the bowl on the plate within a tight tolerance" reflects *Target Pose Accuracy*. These ITCs form the basis for selecting critical motion segments and assigning rewards in our LIBERO-Elegant annotation process. Although defined per task, these ITCs naturally generalize across semantically related manipulation behaviors, enabling consistent evaluation beyond the curated eight tasks.

Table 4 summarizes this mapping from each task's behavioral requirement to its corresponding ITC and primary quality dimension. Figure 6 further visualizes elegant versus non-elegant executions, highlighting where ITC violations occur along the trajectory.

**Operationalizing ITCs in the LIBERO Simulator.** In the original LIBERO setup, task success is specified using the Behavior Domain Definition Language (BDDL), which provides predicates such as `In`, `On`, `Open`, `Close`, `TurnOn`, and `TurnOff` to check whether the final goal conditions are satisfied. However, these predicates encode only *what* outcome is achieved, without constraining *how* the object is manipulated throughout the trajectory.

To enforce Implicit Task Constraints (ITCs), we extend the BDDL predicate set with additional semantics that ground the Elegance Criteria in simulation:

- `IsGrasping`: ensures that the manipulated object remains securely grasped until the intended release moment.
- `IsOnBottomOf`: verifies that the object is stably supported by the target surface without premature dropping.
- `IsPreciselyOn`: enforces accurate placement within a tight positional tolerance.
- `IsOrientationAligned`: checks rotational correctness required for stable insertion or placement.
- `PositionUnchanged`: confirms that nearby objects are not unintentionally disturbed by the motion.

These predicates are incorporated into the BDDL goals of each task to produce an ITC-aware evaluation rule. For instance, Task 0 requires the ketchup bottle not only to be `In` the basket region, but also `IsOnBottomOf` the basket and still `IsGrasping` at release, preventing premature dropping. Tasks 2–3 enforce precise placement via `IsPreciselyOn`, and Tasks 6–7 penalize unintended contact through `PositionUnchanged`.

By embedding ITCs directly into the simulator, LIBERO-Elegant provides automated, consistent evaluation of execution quality: succeeding at the task is necessary but *not* sufficient, the execution must also satisfy the implicit quality constraints that define elegant manipulation.

### B.3. Annotation Protocol and Reward Design

**Motivation.** While the LIBERO-Elegant benchmark defines the qualitative dimensions of manipulation quality, its effective use for learning requires explicit supervision at decision-critical moments. Rather than relying solely on sparse, trajectory-level success signals, we adopt a *task-specific, segment-level annotation scheme* to capture execution quality with respect to Implicit Task Constraints (ITCs).

**Annotation Procedure.** For each demonstration, we manually identify one or more short temporal segments where the ITCs are most relevant, and assign **binary rewards** $r_t \in \{0, 1\}$ indicating whether the constraint is satisfied. For example, in `Task-5` (*place the book in the right compartment of the caddy*), we focus on the final alignment phase: trajectories with precise, controlled placement receive 1, while premature release or misalignment yields 0. Similarly, in `Task-6` (*push the plate to the front of the stove*), reward is given only when the robot avoids contacting surrounding objects during navigation.

**Annotation Tools.** To facilitate this process and ensure consistency, we developed two custom visual tools (Figure 7). The **Elegance Segment Annotator (ESA)** (Figure 7(a)) enables annotators to load a trajectory, inspect multi-view video streams, and interactively mark short evaluation segments at positions where ITCs apply. The tool stores the resulting $[start, end]$ indices as annotation metadata for later reward assignment. The **Reward Validation Viewer (RVV)** (Figure 7(b)) is used for post-annotation verification, allowing annotators to review the temporal distribution of binary rewards and ensure correct alignment with constraint-critical motion.

**Outcome.** This task-specific supervision yields the *Elegance-Enriched Dataset* $\mathcal{D}_{\text{elegant}}$, incorporating **time-aligned** binary indicators of constraint satisfaction throughout the trajectory. This high-quality data serves as the critical bedrock for training the elegance-aware value estimator in Stage 2.

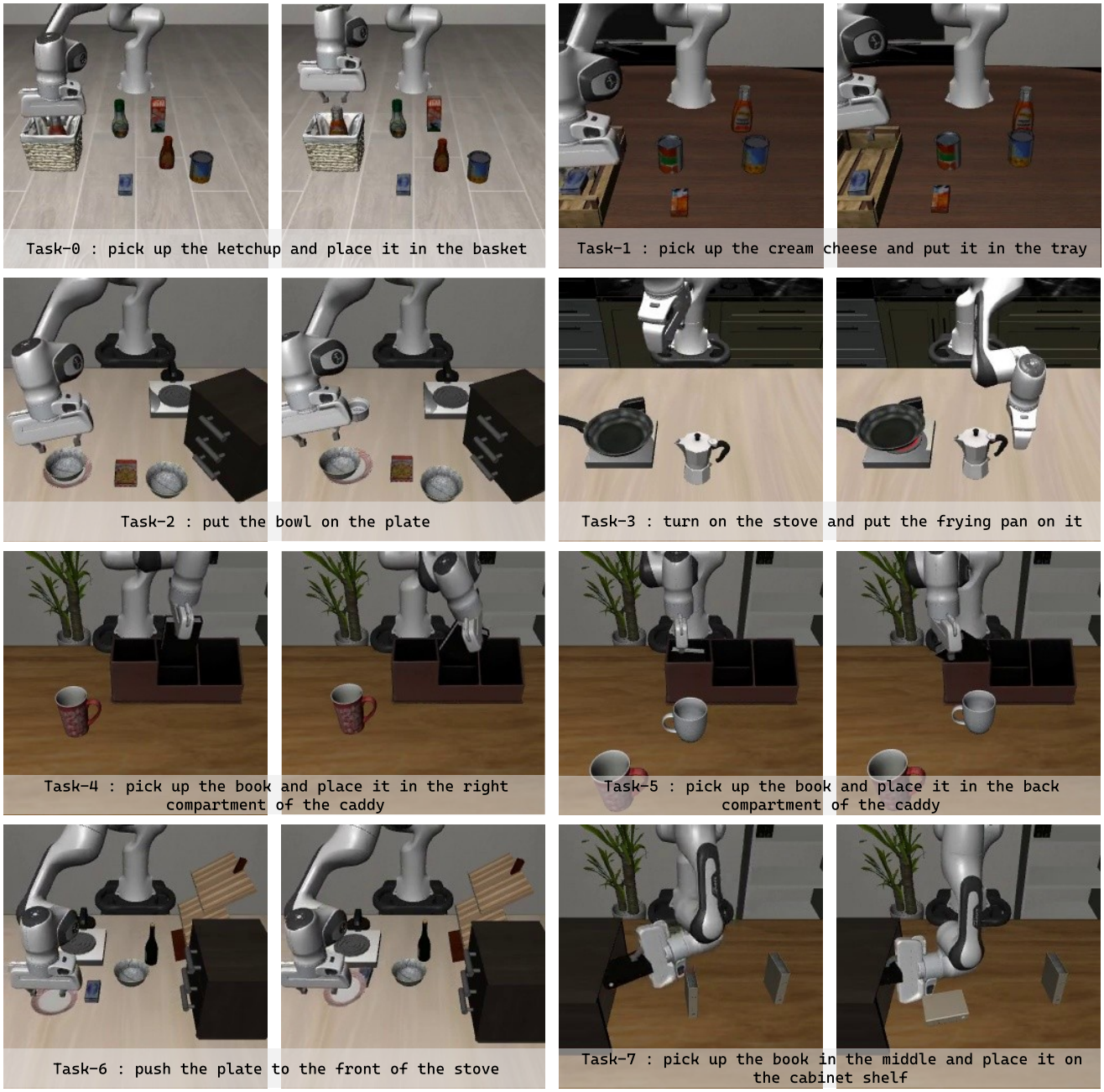



Figure 6. **Visual examples of the LIBERO-Elegant benchmark tasks.** For each task, the left image shows a trajectory satisfying the corresponding *Implicit Task Constraint (ITC)*, while the right image illustrates a non-elegant execution violating that constraint (e.g., early release, misalignment, or unintended collision). This visual contrast highlights the quality dimension emphasized by each **Elegance Criteria**.

## C. Implementation Details for RQ1

**Overview.** To ensure a fair and controlled comparison of execution quality under JITI refinement, we fine-tune three representative base policies on the LIBERO-Elegant dataset: $\pi_{0.5}$ [10], GR00T-N1/N1.5 [27], and SmolVLA-500M [30]. All models share the same observation modalities, embodiment configuration, and data formats, differing only in model capacity and trainable components.

We then integrate our Elegance Critic with these base policies during inference to obtain the *JITI-guided* variants evaluated in the main results. No additional training is ap-

Table 4. **Task list for the LIBERO-Elegant benchmark.** Each task inherits a base instruction from LIBERO but introduces a specific *Implicit Task Constraint (ITC)* that targets one of four **Elegance Criteria**: Task Sequence Integrity, Target Pose Accuracy, Pose Alignment, or Collision-Free Execution.

| Task ID | Instruction | Implicit Task Constraint (ITC) | Elegance Criteria |
|---|---|---|---|
| Task 0 | pick up the ketchup and place it in the basket | The object must remain securely grasped and only be released once fully inside the basket, ensuring no premature drop. | Task Sequence Integrity |
| Task 1 | pick up the cream cheese and put it in the tray | The gripper must retain the object until it is stably placed in the tray; early release constitutes a failure. | Task Sequence Integrity |
| Task 2 | put the bowl on the plate | The bowl must be placed precisely at the plate's center, maintaining positional and rotational accuracy. | Target Pose Accuracy |
| Task 3 | turn on the stove and put the frying pan on it | The frying pan must be centered on the stovetop within a small positional tolerance. | Target Pose Accuracy |
| Task 4 | pick up the book and place it in the back compartment of the caddy | The book must be inserted with correct orientation and full depth into the caddy compartment. | Pose Alignment |
| Task 5 | pick up the book and place it in the right compartment of the caddy | The book must be aligned with the caddy slot and inserted without angular deviation. | Pose Alignment |
| Task 6 | push the plate to the front of the stove | The arm must maintain a smooth trajectory while avoiding any collision with other objects. | Collision-Free Execution |
| Task 7 | pick up the book in the middle and place it on the cabinet shelf | The arm must move the book safely without contacting neighboring objects during transport. | Collision-Free Execution |

plied to the base policies in this stage.

Table 5 summarizes training steps, batch sizes, and compute requirements for the three base policies. Detailed implementation configurations for both the base policies and their JITI-guided versions are provided in the following subsections.

## C.1. Training Configuration for $\pi_{0.5}$

**Model Architecture.** The $\pi_{0.5}$ [10] model is based on a hybrid vision–language–action architecture. We fine-tune a **Paligemma-2B** backbone with LoRA adapters for visual–language encoding, while a **Gemma-300M** head predicts 10-step action sequences per decision step. Only the LoRA-modified modules and action head are trainable; the rest of the backbone remains frozen.

**Dataset and Observations.** Training uses the curated **LIBERO-Elegant** subset in the LeRobotLiberoDataConfig format. Each sample provides synchronized two-view RGB observations (front and wrist cameras), 8D proprioceptive state, the language instruction prompt, and the corresponding 10-step continuous action sequence. All base policies share identical observation modalities to ensure controlled comparisons.

**Optimization Setup.** We train for **30k** steps with a batch size of **8**, using the AdamW optimizer with a gradient clipping norm of **1.0**. The learning rate follows a cosine decay schedule from a peak value of **5e-5**, with a warm-up of **10k** steps. EMA is **disabled**, and weights are initialized from the official `pi05_base` checkpoint.

**Compute and Runtime.** All experiments use a single NVIDIA RTX 5090 (32GB) with `bfloat16` precision. A full run completes in approximately **16 hours**. Checkpoints are saved every **10k** steps.

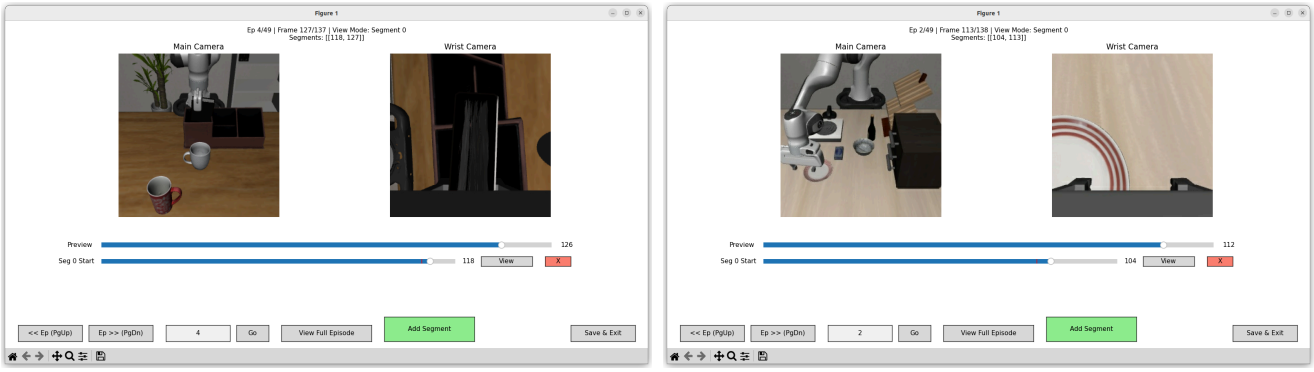
## C.2. Training Configuration for GR00T-N1 and GR00T-N1.5

**Model Architecture.** We fine-tune the GR00T-N1 (2B) [27] and GR00T-N1.5 (3B) [27] models to serve as additional base policies. For both variants, the *vision and language backbones remain frozen*. Only the projection layers and the diffusion-based action policy head are updated. LoRA tuning is disabled.

**Dataset and Observations.** The models are trained on LIBERO-Elegant using the `libero_panda_gripper` embodiment configuration, with identical observation modalities as in $\pi_{0.5}$: two-view RGB images (front and wrist cameras), 8D proprioceptive inputs, language instruction prompts, and the corresponding continuous action sequences.

**Optimization Setup.** Both experiments follow the official GR00T training framework, with customized hyperparameters adapted for the LIBERO-Elegant benchmark. Each model is trained for **50k** steps with a batch size of **16**. We use AdamW with a learning rate of **1e-4**, a warm-up ratio of **0.05**, and a weight decay of **1e-5**. Checkpoints are written every **5k** steps.

(a) Elegance Segment Annotator




(b) Reward Validation Viewer

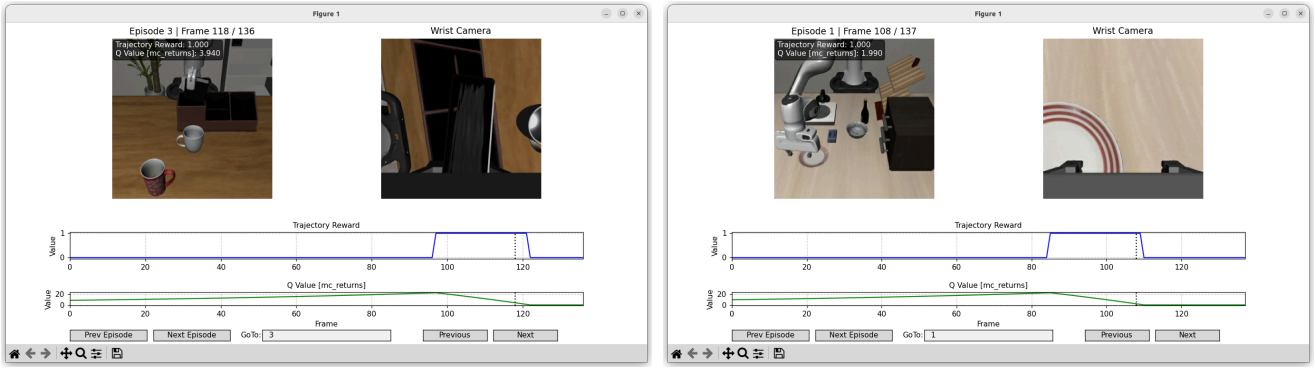



Figure 7. **Overview of the custom tools developed for our elegance annotation workflow**, exemplified using `Task-5` (placing the book) and `Task-6` (pushing the plate). (a) The **Elegance Segment Annotator (ESA)** enables annotators to interactively select key-motion segments where implicit task constraints (ITCs) are evaluated. (b) The **Reward Validation Viewer (RVV)** visualizes the annotated per-frame rewards and the resulting Monte Carlo returns, allowing rapid validation of reward quality and internal consistency.

**Compute and Runtime.** Training is performed on **4**× **NVIDIA A40** GPUs with `bfloat16` precision. A full run takes approximately **40 hours** to complete 50k steps.

## C.3. Training Configuration for SmolVLA

**Model Architecture.** We fine-tuned the **SmolVLA (500M)** [30] model on the **LIBERO-Elegant** dataset to serve as the base policy for our JITI-guided refinement experiments. The model was trained using the official `lerobot` policy training framework, with a configuration adapted for fine-grained manipulation quality evaluation.

**Dataset and Observations.** Training uses the LIBERO-Elegant dataset with the same observation configuration as other base policies: two-view RGB images (front and wrist cameras), 8D proprioceptive inputs, the language instruction prompt, and the corresponding ground-truth action sequence.

**Optimization Setup.** We train for **100k** steps with a batch size of **64**, using AdamW ($\beta_1 = 0.9, \beta_2 = 0.95, \epsilon = 10^{-8}$) and a learning rate of **1e-4**. The LR follows a cosine decay schedule with **1k** warm-up steps. Mixed precision is disabled for stability. Checkpoints are saved every **25k** steps, and evaluations are conducted every **20k** steps.

**Compute and Runtime.** Training is performed using a single NVIDIA RTX 5090 (32GB GPU). A full run takes approximately **17 hours**.

Table 5. **Training summary of base policies evaluated in RQ1.**

| Model | Steps | Batch | Compute (GPU) | Time |
|---|---|---|---|---|
| $\pi_{0.5}$ | 30k | 8 | RTX 5090 (1) | 16 h |
| GR00T-N1 | 50k | 16 | NVIDIA A40 (4) | 40 h |
| GR00T-N1.5 | 50k | 16 | NVIDIA A40 (4) | 40 h |
| SmolVLA-500M | 100k | 64 | RTX 5090 (1) | 17 h |

## C.4. Elegance Critic Training Details

**Training with SmolVLA Features.** For the SmolVLA-based setup, the Elegance Critic is trained on top of *frozen* SmolVLA features. The backbone processes two-view RGB observations, proprioceptive states, and the language instruction, and we project its hidden representation into the critic backbone space. The critic predicts values over

action chunks of length $K = 10$ in the LIBERO simulation environment, with a matching temporal-difference offset offset $= 10$. We train for 20k gradient steps with a batch size of 32, using a soft target-network update rate $\rho = 5.0 \times 10^{-3}$.

The optimization setup uses separate learning rates for different components: the actor and critic heads use $\text{LR}_\pi = \text{LR}_Q = 1.0 \times 10^{-5}$, the visual–language backbone and projection layers use $\text{LR}_{\text{VLM}} = 3.0 \times 10^{-6}$, the temperature parameter $\tau$ uses $\text{LR}_\tau = 2.0 \times 10^{-5}$, and the CQL [12] regularization weight $\alpha$ is updated with $\text{LR}_\alpha = 1.0 \times 10^{-4}$. We adopt a fixed CQL coefficient $\alpha = 5.0$ without autotuning, with a target action gap of 0.5. All models are trained in `bfloat16` with gradient checkpointing enabled. The critic backbone is based on `SmolVLM2-256M-Video-Instruct` [21], fed by a linear projection from the SmolVLA feature dimension to its hidden size.

**Training with GR00T-N1.5 Features.** For the GR00T-based setup, we follow a similar Cal-QL [25] training scheme, but use frozen GR00T-N1.5 features as input. GR00T-N1.5 encodes the same observation tuple, and all GR00T components remain frozen during critic training. The critic operates on action chunks of length $K = 10$ with base action dimension $D_{\text{act}} = 7$. We use $\gamma = 0.98$, batch size 32, and train for 20k gradient steps, with gradient norm clipping at 0.5 and the same soft update rate $\rho = 5.0 \times 10^{-3}$.

The learning rate configuration mirrors the SmolVLA case, except for a lower actor learning rate $\text{LR}_\pi = 1.0 \times 10^{-6}$ to stabilize training under the diffusion-based action space. We again use a fixed CQL coefficient $\alpha = 5.0$ without autotuning, with a target action gap of 0.5 and an (unused) Lagrange multiplier initialized at 0.1. The critic backbone here is `nvidia/Eagle2-1B` [15], with a linear projection from the GR00T feature dimension to the corresponding hidden size, and all components are trained in `bfloat16` precision.

# D. Task Specification for Generalization (RQ3)

**Overview.** This section provides detailed task definitions and configurations used in the generalization study (RQ3), where the Elegance Critic trained on a limited set of tasks is evaluated on unseen but semantically related tasks.

## D.1. Generalization Setup: Seen and Unseen Tasks

To assess the critic's ability to generalize its learned notion of elegance, we evaluate it on a family of pick-and-place tasks divided into two groups, as illustrated in Figure 8. Specifically:

- **Seen Tasks:** Three tasks used for critic training, placing everyday objects such as *milk*, *ketchup*, and *alphabet soup* into a basket. These tasks share identical success conditions but differ in spatial layouts, object geometry and visual appearance. Additionally, these tasks require that the object remain securely grasped throughout the entire motion and only be released once fully inside the basket, ensuring no premature drops.
- **Unseen Tasks:** Four novel but semantically similar tasks, placing *salad dressing*, *BBQ sauce*, *tomato sauce*, and *orange juice* into the same basket. These tasks are excluded during training to test the critic's ability to generalize to unseen object appearances and spatial layouts. Similar success conditions apply, enforcing secure grasping and proper release timing to meet criteria for elegant task execution.

## D.2. Task Specifications and Quantitative Analysis

Table 6 provides detailed specifications of these tasks, including instructions and elegance criteria. Additionally, quantitative evaluations in the main paper show substantial ESR gains on both seen and unseen tasks, demonstrating strong transferability.

# E. Real-World Experiment Suite

**Overview.** We validated our approach on a physical robotic system to assess its real-world practicality. These tests evaluate whether the Elegance Critic can effectively enhance policy execution under real-world challenges such as sensor noise, actuation delays, and object variability, conditions that are difficult to replicate in simulation.

## E.1. Experimental Setup

**Hardware.** Our experimental setup consists of a **SO100** arm equipped with its native gripper. Visual observations are captured by two cameras : a static front camera and a wrist-mounted camera, both providing 640×480 RGB data. All policy inference and control commands are processed on a desktop workstation with an **NVIDIA RTX 5090 GPU**.

## E.2. Real-World Task Suite

We designed a suite of 6 real-world manipulation tasks, derived from common household activities. As detailed in Table 7, each task is defined not only by its base instruction (e.g., "put the bowl on the plate") but by a corresponding *Implicit Task Constraint (ITC)* that specifies the quality of the execution.

To visually illustrate these constraints, Figure 9 presents a side-by-side comparison for each task, contrasting a successful, elegant execution against a non-elegant failure case. **Software and Control** The system is built entirely within the **LeRobot software framework**. We implement our JITI-guided **SmolVLA** policy within this framework. The control loop runs at **30 Hz**. At each timestep, the system receives RGB observations from both cameras, the JITI

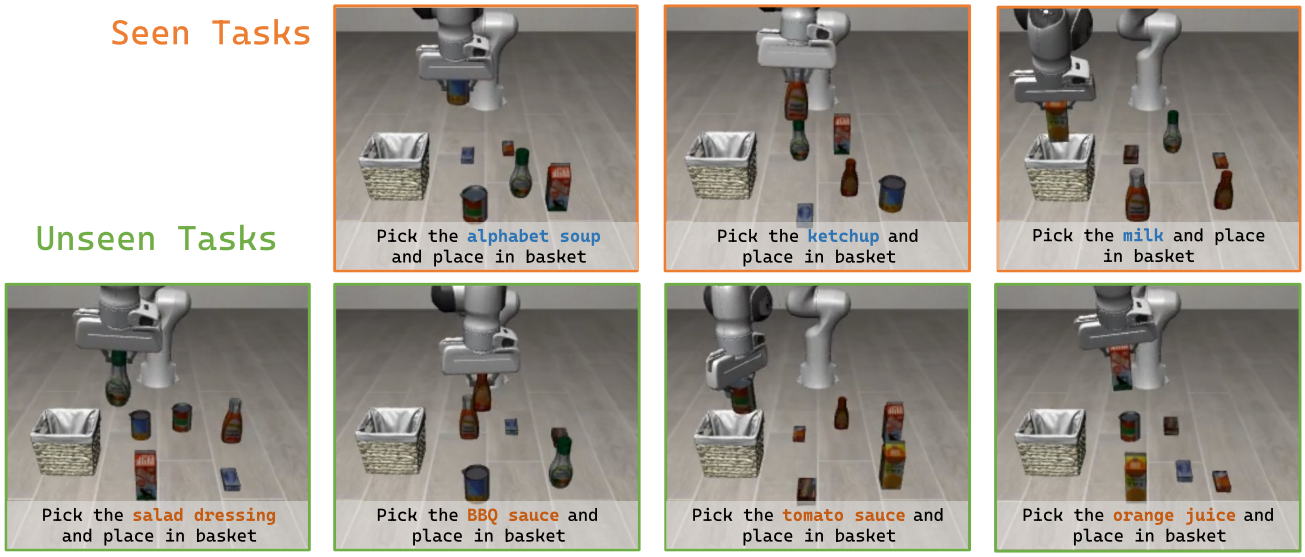



Figure 8. **Visualization of tasks used in the generalization study.** The top row shows the three **Seen Tasks** (*alphabet soup*, *ketchup*, and *milk*) used for training the Elegance Critic, while the bottom row shows the four **Unseen Tasks** (*salad dressing*, *BBQ sauce*, *tomato sauce*, and *orange juice*) used for evaluation without fine-tuning. All tasks share identical success conditions but differ in spatial layouts, object geometry, and visual appearance.

framework triggers intervention if necessary, and the resulting action dictionary is sent to the robot's low-level controller.

## F. Extended Qualitative Analysis

**Overview.** In this section, we provide a detailed qualitative case study to illustrate the fine-grained decision-making of our model.

### F.1. Case Study: JITI-Guided Elegance Refinement

To further illustrate how JITI refines actions at decision-critical states, we present a qualitative case study conducted on a shelf-placement task.

At a decision-critical moment, the Elegance Critic evaluates eight candidate action sequences sampled from the base VLA policy. We then branch the rollout into three distinct trajectories: one that always selects the **Best-Q** candidate with the highest predicted elegance, another that selects the **Median-Q** candidate ranked in the middle, and a third that selects the **Worst-Q** candidate with the lowest predicted elegance.

Each trajectory continues to apply the same selection rule at every JITI intervention trigger, always selecting the highest, middle, or lowest-ranked candidate respectively. This causes the three behaviors to diverge over time, demonstrating how persistent high- or low-quality decisions ultimately translate into substantial differences in execution elegance.

As visualized in Figure 10, the Best-Q trajectory maintains smooth, collision-free motion and achieves precise placement of the book into the target compartment with correct pose alignment. The Median-Q trajectory incurs a slight edge contact and a small amount of tilt, but still results in successful placement. In contrast, the Worst-Q trajectory suffers from repeated collisions with the shelf and severely degraded object orientation, despite eventually completing the task.

## G. Additional Experimental Results

### G.1. Automated Discovery and Scalability of ITCs

We present a training-free pipeline based on *Cosmos-Reason2* that automatically discovers ITCs from task descriptions and video observations. We then infer ITCs for all **73 LIBERO** tasks and test scalability on **150 BridgeData V2** and **50 AgiBot World** tasks, which extend beyond LIBERO-style table-top pick-and-place and reflect more diverse real-world manipulation scenarios. Inferred ITCs were validated by experts, yielding $86.08\%$ accuracy (235/273), demonstrating the potential for generalization across large-scale and diverse manipulation tasks.

### G.2. Scalable Annotation

We further designed a two-stage automatic labeling pipeline based on *Cosmos-Reason2*. Stage one uses the given ITCs to prompt the model to select segments where the constraints are most relevant. And stage two evaluates each ITC on those focused segments to decide whether it is satisfied. Evaluating four LIBERO-Elegant tasks to assess label

Table 6. **Task specification for the generalization study (RQ3).** The Elegance Critic is trained on three **Seen Tasks** and evaluated directly on four **Unseen Tasks** without fine-tuning. All tasks share the same pick-and-place semantics but differ in object.

| Category | Object | Instruction | Elegance Criteria |
|---|---|---|---|
| **Seen Tasks (Training)** | | | |
| Seen-A | *milk* | pick up the milk and place it in the basket | Task Sequence Integrity |
| Seen-B | *ketchup* | pick up the ketchup and place it in the basket | Task Sequence Integrity |
| Seen-C | *alphabet soup* | pick up the alphabet soup and place it in the basket | Task Sequence Integrity |
| **Unseen Tasks (Evaluation)** | | | |
| Unseen-A | *salad dressing* | pick up the salad dressing and place it in the basket | Task Sequence Integrity |
| Unseen-B | *BBQ sauce* | pick up the BBQ sauce and place it in the basket | Task Sequence Integrity |
| Unseen-C | *tomato sauce* | pick up the tomato sauce and place it in the basket | Task Sequence Integrity |
| Unseen-D | *orange juice* | pick up the orange juice and place it in the basket | Task Sequence Integrity |

Table 7. **Task list for the real-world evaluation suite.** Each task inherits a base instruction but introduces a concrete *Implicit Task Constraint (ITC)* that emphasizes motion quality and defines the corresponding Elegance Criteria.

| Task ID | Instruction | Implicit Task Constraint (ITC) | Elegance Criteria |
|---|---|---|---|
| Task 0 | pick up the carrot and put it in the basket | The gripper must hold the carrot firmly and release only after it is fully inside the basket to avoid premature dropping. | Task Sequence Integrity |
| Task 1 | pick up the corn and put it on the plate | The corn must remain stably grasped and released only when securely positioned on the plate, avoiding free fall. | Task Sequence Integrity |
| Task 2 | stack the red squares on top of the green ones | The red block must be placed precisely aligned with the green one, maintaining both positional and rotational accuracy. | Target Pose Accuracy |
| Task 3 | stack the blue squares on top of the yellow ones | The blue block must be placed directly above the yellow block, ensuring stable and centered stacking. | Target Pose Accuracy |
| Task 4 | put the bowl on the plate | The bowl must be centered precisely on the plate with minimal offset or tilt. | Target Pose Accuracy |
| Task 5 | close the top drawer of the cabinet | The drawer must be closed smoothly and fully until it is tightly shut. | Pose Alignment |

quality, we observed that stage one's segment selection was nearly always correct. In stage two, the VLM stays reliable on process-oriented constraints with $70.6\%$ accuracy, but it struggles with finer spatial checks, with $47.8\%$ accuracy, reflecting current spatial-reasoning limits. Even so, the approach sharply reduces human labeling effort and keeps the path to scale open.

## G.3. Comparison against Simpler Baselines.

We evaluated simpler baselines that use the trained critic for static data reshaping: *Filtered Data* and *Weighted Data*[36]. As shown in Table 8, our method consistently outperforms these static alternatives.

Table 8. Comparison with static baselines.

| Method | T-0 | T-1 | T-2 | T-3 | T-4 | T-5 | T-6 | T-7 | Avg. |
|---|---|---|---|---|---|---|---|---|---|
| SmolVLA (Base) | 70 | 30 | 34 | 52 | 50 | 42 | 48 | 72 | 49.8 |
| + Filtered Data | 58 | 32 | 30 | 38 | 46 | 48 | 46 | 86 | 48.0 |
| + Weighted Data | 64 | 40 | 38 | 40 | 54 | 36 | 42 | 74 | 48.5 |
| **Ours (JITI) + SmolVLA** | **86** | **68** | **42** | **62** | **60** | **54** | **74** | **92** | **67.2** |

## G.4. Generalization Scope.

We expanded RQ3 to address generalization scope, as shown in Table 9. For rigid objects, our method demonstrates robust transfer to unseen objects with distinct geometries (e.g., from milk to bowl). Regarding articulated

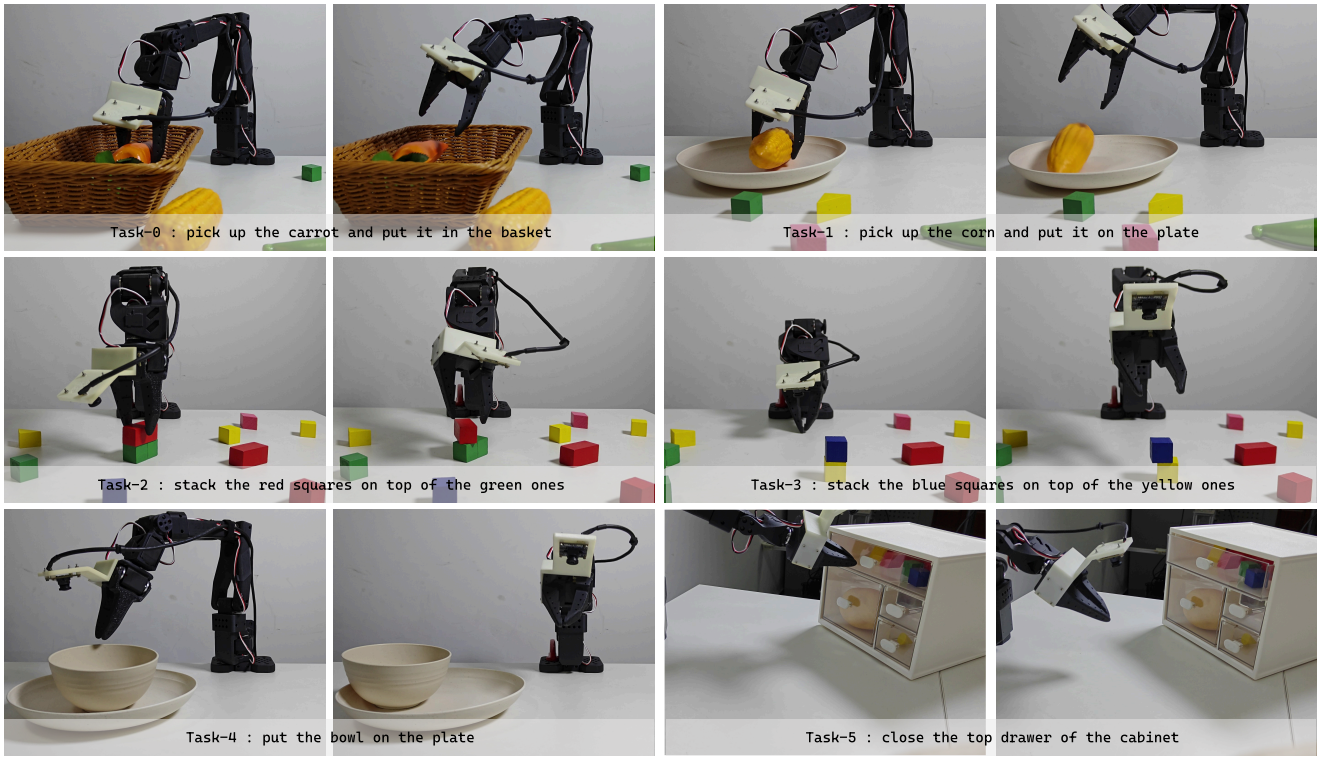



Figure 9. **Visual overview of the real-world task suite (defined in Table 7).** For each task, we present a side-by-side comparison: (**Left**) An elegant execution that successfully satisfies the task's *Implicit Task Constraint (ITC)*. (**Right**) A non-elegant execution that violates the constraint, resulting in failures such as premature dropping, poor alignment, or incomplete motion. This comparison visualizes the fine-grained **Elegance Criteria** we evaluate.

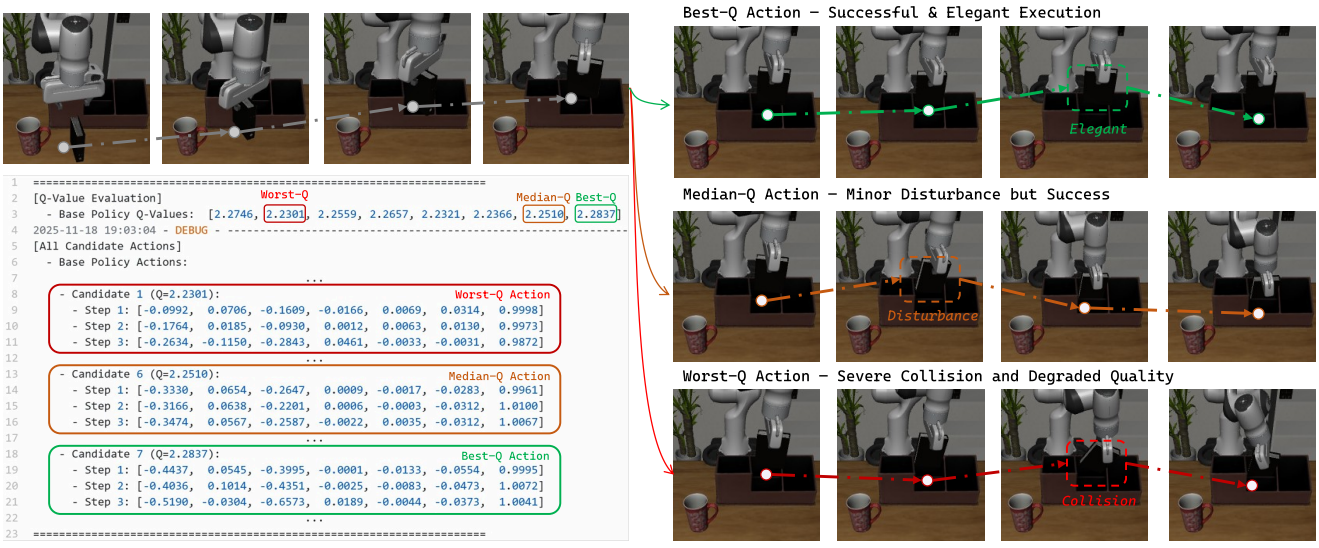



Figure 10. **Effect of JITI on execution elegance at a decision-critical moment.** We visualize three rollout branches using Best-Q, Median-Q, and Worst-Q intervention strategies at every JITI trigger. Higher predicted Q-values result in smoother, collision-free execution and correct pose alignment, whereas repeated low-Q selection leads to severe collisions and degraded object orientation.

objects, we trained on drawer-closing tasks and evaluated transfer to unseen drawers as well as objects with different articulation types, e.g., closing a microwave. In both cases, our approach consistently outperforms the base

model, demonstrating strong generalization.

Table 9. Generalization on diverse rigid and articulated objects.

| **Unseen Rigid Objects** | | | | |
|---|---|---|---|---|
| **Method** | Ketchup | Cream Cheese | Black Bowl | Avg. |
| SmolVLA (Base) | 34 | 54 | 20 | 36.0 |
| **Ours (JITI) + SmolVLA** | **56** | **68** | **46** | **56.7** |
| **Unseen Articulated Objects** | | | | |
| **Method** | Close Drawer | Close Microwave | – | Avg. |
| SmolVLA (Base) | 30 | 40 | – | 35.0 |
| **Ours (JITI) + SmolVLA** | **64** | **50** | – | **57.0** |

## G.5. Robustness of Q-value Heuristic.

We performed a sensitivity analysis by varying the threshold from $0.01$ to $10$ times the default value. Results show a broad stable range, alleviating brittleness concerns. ESR is $64.86\%$ at $0.5\times$, $65.72\%$ at $1\times$, and $62.86\%$ at $2\times$. Performance drops only at extremes: $52.86\%$ at $0.01\times$ (over-sensitive) and $48.86\%$ at $10\times$ (under-sensitive), confirming the heuristic reliably captures critical moments and is robust.

## G.6. Computational Overhead Analysis.

We measured average inference time per step over 50 episodes: No Guidance 112.4 ms, Full Guidance 449.3 ms, Proposed Method 220.5 ms. With 33.9% intervention and an action chunk of 10, our method achieves an effective control frequency of approximately 45 Hz, supporting real-time operation.