ATHENA: Accelerated Multi-Task Heterogeneous Influence Functions for Robot Data Curation

Tao Xu[1,2], **Jiaxin Wang**[2,3], **Runhao Zhang**[2], **Jiayi Guan**[1], **Xianchao Zeng**[2], **Weixi Song**[2], **Xinyu Zhou**[2], **Zhetao Chen**[2], **Guang Chen**[1,2], **Yong-Lu Li**[2,4,†]

[1]Tongji University, [2]Shanghai Innovation Institute, [3]Xi'an Jiaotong University, [4]Shanghai Jiao Tong University

Abstract: In robot imitation learning, influence functions provide a principled approach to quantify each demonstration's effect on robot task outcomes, yet scaling them to billion-parameter Vision-Language-Action (VLA) models is limited by computational and multitask bottlenecks. To this end, we propose ATHENA, an influence function framework tailored for multitask VLA data curation at a billion-parameter scale. Concretely, it leverages the Kronecker structure of linear-layer gradients to reduce projection cost, and approximates dense Hessian inversion with a rank-r Random Truncated Approximation, achieving about a 313.4$\times$ speedup in influence computation. Furthermore, ATHENA formulates global and local interactive influence to balance data curation across 50 jointly trained tasks. Extensive evaluations on RoboTwin 2.0 and real-robot deployment, covering 9.34 and 6.90 hours of demonstrations, respectively, show that ATHENA matches or exceeds full-data joint fine-tuning using only 50% of demonstrations in simulation and 66.7% of data across six real-robot tasks. Overall, ATHENA demonstrates its effectiveness for data curation in billion-parameter multitask VLA fine-tuning. Project website: this URL.

Keywords: Imitation learning, Data Curation, Influence Functions

1 Introduction

Vision-Language-Action (VLA) models have shown promising results in robotic manipulation from large-scale robot demonstrations [1, 2, 3], but their performance also depends critically on data quality [4, 5, 6]. Naively scaling demonstration data for fine-tuning can incur high costs while yielding limited or even degraded performance [7, 8]. This motivates a central question for VLA fine-tuning: *which demonstrations should be retained to maximize the performance of VLA models?*

Answering this question requires efficient, data-centric methods to identify high-quality subsets of fine-tuning demonstrations. However, existing approaches face challenges when applied to billion-parameter VLA models. Specifically, traditional leave-one-out data valuation requires retraining the models for each candidate subset, which incurs prohibitive computational cost [9]. Expert-based or data distillation curation methods are more efficient, but they often do not capture the causal effect of training data on downstream policy performance [5, 10, 11]. In contrast, gradient-based influence functions [12] provide a principled and interpretable framework without repeated retraining. For example, CUPID [8] applies influence functions to estimate the effect of each robot demonstration, curating high-quality subsets that match or exceed full-data training.

However, CUPID is limited to small imitation policies (24M parameters) and single-task curation [8, 13], preventing direct scaling to billion-parameter multitask VLA models such as π_0 [3]. Specifically, per-sample gradient computation incurs $\mathcal{O}(DP)$ cost, where D is the number of model parameters and P is the projection dimension, and dense Hessian inversion entails $\mathcal{O}(NP^2 + P^3)$ complexity, where N is the number of training samples, which together limit the scalability of influence functions [14, 15, 16]. Moreover, because influence scores are computed greedily [17, 18], naively extending single-task attribution to multitask settings may produce imbalanced curation across tasks [19, 20].

[†]Corresponding author: yonglu.li@sjtu.edu.cn

To overcome these challenges, we introduce ATHENA, an influence function framework for multitask VLA data curation at a billion-parameter scale. Concretely, it improves computational efficiency along two axes: gradient projection at each layer and Hessian approximation. For gradient projection, ATHENA leverages the Kronecker structure of linear layer gradients to perform low-rank projections at each layer, achieving a square root reduction in projection cost at each layer, from $\mathcal{O}(DP)$ to $\mathcal{O}(\sqrt{DP})$. For Hessian approximation, ATHENA approximates dense Hessian inversion with a rank-r Random Truncated Approximation, lowering the leading cost from $\mathcal{O}(N \cdot P^2 + P^3)$ to $\mathcal{O}(N \cdot P \cdot r)$. This reduces influence computation from 8054.6 to 25.7 GPU-hours, corresponding to about a 313.4$\times$ speedup. Furthermore, ATHENA formulates multitask influence interaction with global and local interactive influence measures for balanced curation over 50 jointly trained tasks, improving the return on investment (ROI) of data curation.

Our contributions are summarized as follows: (1) We propose ATHENA, an influence function framework for multitask VLA data curation at billion-parameter scale. It leverages Kronecker-structured gradient projection and a rank-r Random Truncated Approximation to significantly reduce the cost of influence function computation, achieving a $313.4\times$ speedup. (2) We present a global and local influence design that balances multitask data curation and improves ROI across 50 jointly trained tasks. (3) We evaluate ATHENA in RoboTwin 2.0 simulation and real-robot deployment, where it matches or exceeds full-data joint fine-tuning using only 50% of demonstrations in simulation and 66.7% of data across six real-robot tasks. These results demonstrate its effectiveness for VLA fine-tuning data curation.

2 Related Work

Robot Imitation Learning. Imitation learning has become central to robot research [21, 22, 23, 24, 25]. Early methods such as ACT [26] and Diffusion Policy [13] achieve excellent performance under task-specific training, but training one policy per task reduces the ROI of joint robot learning and limits multitask data scaling. In contrast, VLA models leverage large-scale pretrained knowledge and multitask fine-tuning to improve both performance and ROI [27, 28]. However, their performance depends not only on data scale, but also on demonstration quality [29, 30]. Large-scale yet redundant datasets may even harm VLA performance [31, 32], making data curation essential.

Robot Data Curation. Recent robot data curation methods aim to improve policy performance by identifying high-utility demonstrations [33, 34, 35]. Leave-one-out valuation and Data Shapley directly estimate data utility but require repeated retraining [9], which incurs unacceptable computational cost [36]. More efficient methods, including dataset distillation [37, 38], quality scoring [10, 11], retrieval [39, 40, 41, 42], and mixture learning [43, 44], avoid retraining but rely on proxy objectives rather than the causal effects on downstream policy performance. Gradient-based curation methods, including CUPID [8], QoQ [7], and DataMIL [20], estimate demonstration utility for imitation learning, yet scaling them to billion-parameter multitask VLA fine-tuning remains challenging [31, 18]. In practice, as robot imitation learning shifts from small task-specific policies to billion-parameter multitask VLA models, robot data curation also requires an efficient method tailored to VLA fine-tuning.

3 Preliminaries and Problem Formulation

3.1 Robot Data Curation with Influence Functions

We consider a VLA model π_θ, $\theta \in \mathbb{R}^D$, fine-tuned on demonstrations $\mathcal{D} = \{\xi_i\}_{i=1}^n$. Each demonstration trajectory consists of sequential state-action pairs, formulated as individual training samples $z_t^i = (s_t^i, a_t^i)$. Thus, a demonstration is denoted as $\xi_i = \{z_t^i\}_{t=1}^{H_i}$, and the total number of training timesteps is $N = \sum_i H_i$. Let $\theta^\star$ denote the fine-tuned parameters and $\mathcal{L}$ the imitation loss. Classical influence functions [12] estimate the first-order effect of upweighting a single training sample z on a test quantity $f(\hat{z}; \theta)$, with $H_\theta = \nabla_\theta^2 \mathcal{L}(z_t^i; \theta^\star)$:

$$\Psi_{\text{inf}}(\hat{z}, z) = -\nabla_\theta f(\hat{z}; \theta^\star)^\top H_\theta^{-1} \nabla_\theta \mathcal{L}(z; \theta^\star). \tag{1}$$

However, robot demonstrations are sequential trajectories whose influence scores should reflect closed-loop task performance rather than step-wise losses [8, 7]. Accordingly, motivated by CUPID's [8] closed-loop attribution principle, we instantiate the action-level influence $\Psi_{a\text{-inf}}(\hat{z}, z)$ for the flow-matching VLA model π_0, and aggregate it into demonstration-level performance influence:

$$\widehat{\Psi}_{\pi\text{-inf}}(\xi_i) = \frac{1}{m} \sum_{\tau_j \in \mathcal{E}} \frac{R(\tau_j)}{H_i} \sum_{\hat{z} \in \tau_j} \sum_{z \in \xi_i} \Psi_{a\text{-inf}}(\hat{z}, z), \tag{2}$$

where $\mathcal{E}$ contains m evaluation rollouts, $R(\tau_j) \in \{1, -1\}$ is the return of rollout τ_j, H_i is the length of demonstration ξ_i, and $\Psi_{a\text{-inf}}$ is mathematically derived in Appendix A.1.

3.2 Scalability Barriers for Billion-Parameter VLA Curation

Eq. (2) defines a closed-loop performance influence function for robot data curation. However, scaling it to billion-parameter multitask VLA models faces two main bottlenecks: costly influence estimation at the model parameter and data scale, and imbalanced greedy selection under heterogeneous task distributions.

Scaling bottleneck. Naively computing projected gradients requires materializing $g_i = \nabla_\theta \ell(z_i; \theta^\star) \in \mathbb{R}^D$ and projecting it into a P-dimensional feature space. Let $\Omega \in \mathbb{R}^{D \times P}$ denote the random projection matrix. The projected gradient feature is:

$$\phi_i = \Omega^\top g_i. \tag{3}$$

The vector $\phi_i \in \mathbb{R}^P$ denotes the projected feature of z_i. This incurs $\mathcal{O}(DP)$ gradient computation and storage cost, which is prohibitive for billion-parameter VLA models [15, 45, 46].

Subsequently, training gradients are stacked into a projected gradient matrix $G \in \mathbb{R}^{N \times P}$, whose i-th row is $\phi_i^\top$. Influence scores are computed via a damped Gauss-Newton approximation [14]:

$$\widehat{\psi}(z_{\text{te}}, z_{\text{tr}}) = \phi_{\text{te}}^\top \left(G^\top G + \lambda I_P\right)^{-1} \phi_{\text{tr}}. \tag{4}$$

In Eq. (4), ϕ_{tr} and ϕ_{te} denote the projected gradients of training and evaluation timesteps, λ is the damping coefficient, and I_P is the identity matrix. Although this formulation avoids forming the full $D \times D$ Hessian, it still introduces dense computation over the projected training matrix: forming and inverting $G^\top G + \lambda I_P$ costs $\mathcal{O}(NP^2 + P^3)$ [47, 48, 49], which becomes a severe computational bottleneck as N increases. Therefore, billion-parameter VLA curation requires reducing both projected gradient computation and projected inverse-Hessian approximation costs.

Multitask curation bottleneck. For curation over K tasks, estimating influence and fine-tuning independently for each task increases the total computational and storage cost by a factor of K, yielding low ROI. A single greedy influence ranking shared across all tasks is more efficient, but tends to favor tasks with stronger gradient signals, leading to imbalanced task coverage [18, 19, 20]. Multitask VLA curation therefore requires an attribution rule that accounts for both influence across tasks and relevance within each task.

4 ATHENA: Scalable Influence Function Data Curation for VLA Models

To overcome the scalability and multitask bottlenecks in Section 3, we propose ATHENA, an influence function framework for billion-parameter multitask VLA models. Specifically, ATHENA reduces computational and memory constraints by combining Kronecker compressed per-sample gradient featurization with a Random Truncated Approximation of the inverse Hessian. In addition, ATHENA models both global and local influence interactions, enabling effective multitask curation without greedy task imbalance. Figure 1 presents an overview of the pipeline.

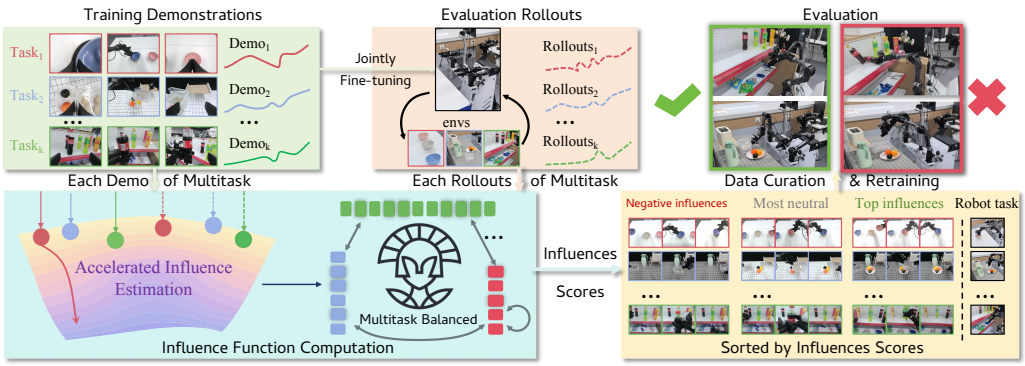


Figure 1: **ATHENA pipeline.** Training dataset and closed-loop rollouts from VLA evaluation are fed into an efficient multitask influence computation module to score and rank demonstration importance, guiding high-quality data curation for VLA fine-tuning.

4.1 Accelerated Influence Estimation

Kronecker Compressed Gradient Featurization. Following Eq. (3), let $g_i = \nabla_\theta \ell(z_i; \theta^\star) \in \mathbb{R}^D$ denote the gradient of the i-th training sample. Naive gradient projection computes $\phi_i = \Omega^\top g_i$ with $\Omega \in \mathbb{R}^{D \times P}$, which costs $\mathcal{O}(DP)$ per sample and $\mathcal{O}(NDP)$ over the training set. In a batched implementation, it must also hold full parameter gradients and the projection matrix, incurring $\mathcal{O}(BD)$ and $\mathcal{O}(DP)$ memory, respectively, for batch size B. This makes projected gradient construction memory-bound for billion-parameter VLA models.

Following LoGRA [15], we avoid explicit gradient projection in parameter space by exploiting the Kronecker structure of linear layer gradients. Specifically, for layer ℓ with activation x_i^ℓ and backpropagated error δ_i^ℓ, the weight gradient admits the outer product form $\delta_i^\ell (x_i^\ell)^\top$. ATHENA projects these two factors before forming the projected feature:

$$\bar{g}_i^\ell = (P_{\text{out}}^\ell)^\top \delta_i^\ell (x_i^\ell)^\top P_{\text{in}}^\ell. \tag{5}$$

The resulting layer features are flattened and concatenated across layers. This replaces dense projection in parameter space with bilateral projection in activation space, reducing the leading projection cost from $\mathcal{O}(DP)$ to $\mathcal{O}(\sqrt{DP})$ while avoiding materialization of full gradients in $\mathbb{R}^D$.

Random Truncated Approximation. After Kronecker compressed featurization, ATHENA stacks the timestep features into $G \in \mathbb{R}^{N \times P}$, where each row is $\phi_i^\top$. Following Eq. (4), a dense damped Gauss–Newton approximation requires forming and inverting $G^\top G + \lambda I_P$, which costs $\mathcal{O}(NP^2 + P^3)$. This cost grows quickly with the number of timesteps, demonstrations, and tasks.

We avoid this dense projected inverse by applying a rank-r Random Truncated Approximation (RTA) [50, 51] to G. Concretely, ATHENA approximates the compressed gradient matrix by its leading randomized spectral components, $G \approx U_r \Sigma_r V_r^\top$, where $V_r \in \mathbb{R}^{P \times r}$ and $r \ll P$. Since $G^\top G \approx V_r \Sigma_r^2 V_r^\top$, ATHENA projects each timestep feature into the retained subspace as $\tilde{\phi}_i = V_r^\top \phi_i$, and computes:

$$\widehat{\psi}_{\text{RTA}}(z_{\text{te}}, z_{\text{tr}}) = \tilde{\phi}_{\text{te}}^\top (\Sigma_r^2 + \lambda I_r)^{-1} \tilde{\phi}_{\text{tr}}. \tag{6}$$

This reduces the leading data-dependent cost from $\mathcal{O}(NP^2)$ to $\mathcal{O}(NPr)$, while replacing a P-dimensional dense inverse with an r-dimensional damped inverse. Additional details on Kronecker compressed featurization and RTA are provided in Appendix A.2.

4.2 Interactive Influence for Multitask Scaling

In multitask VLA curation, each demonstration may affect its own task differently from other tasks. To capture these interactions, we propose Multitask Influence Interaction (MII), which models both

the influence on the demonstration's own task and other tasks to enable balanced multitask data curation. Let demonstration i belong to task $c(i) \in \{1, \ldots, K\}$, with evaluation rollouts $\mathcal{E}$. We define task-local and cross-task influence as:

$$\widetilde{\Psi}^{c(i)}_{\pi\text{-inf}}(i) = \frac{1}{H_i} \sum_{\tau_j \in \mathcal{E}_{c(i)}} R(\tau_j) S_{j,i}, \quad \widetilde{\Psi}^{\text{all}-c(i)}_{\pi\text{-inf}}(i) = \frac{1}{H_i} \sum_{\tau_j \in \mathcal{E} \setminus \mathcal{E}_{c(i)}} R(\tau_j) S_{j,i}, \tag{7}$$

where $S_{j,i}$ denotes the contribution of demonstration i to rollout τ_j (aggregated over timestep pairs). The first term measures local influence, while the second captures cross-task effects.

Raw values of $\widetilde{\Psi}^{c(i)}_{\pi\text{-inf}}$ and $\widetilde{\Psi}^{\text{all}-c(i)}_{\pi\text{-inf}}$ are defined on separate rollout sets and have different numerical scales. We convert each component to a normalized, sorted score to combine local and cross-task influence consistently in the MII score.

$$r_i^{c(i)} = \text{rank}_{\downarrow}^{c(i)}\big(\widetilde{\Psi}^{c(i)}_{\pi\text{-inf}}(i)\big), \quad r_i^{\text{all}-c(i)} = \text{rank}_{\downarrow}^{\text{all}-c(i)}\big(\widetilde{\Psi}^{\text{all}-c(i)}_{\pi\text{-inf}}(i)\big), \tag{8}$$

and define normalized utilities:

$$u_i^{c(i)} = \max(\varepsilon, 1 - r_i^{c(i)}/n_{c(i)}), \quad u_i^{\text{all}-c(i)} = \max(\varepsilon, 1 - r_i^{\text{all}-c(i)}/(n - n_{c(i)})), \tag{9}$$

where $n_{c(i)}$ is the number of demonstrations in task $c(i)$, n is the total number of demonstrations, and $\varepsilon > 0$ is a small numerical floor.

Proposition 4.1 (MII)**.** *The cross-task influence function for curating balanced subsets of demonstrations from multitask datasets is mathematically formulated as:*

$$f_i^{\text{MII}} = u_i^{c(i)} \cdot u_i^{\text{all}-c(i)}. \tag{10}$$

Training demonstrations are sorted by f_i^{MII} to curate a balanced subset that preserves locally critical examples while accounting for cross-task interactions.

The details of the multitask estimation procedure are summarized in Appendix A.3.

5 Experiments

5.1 Experimental Setup

We evaluate ATHENA on RoboTwin 2.0 [6] and six real robot manipulation tasks, using 50K training steps as in CUPID [8], comparing against Oracle [11], Random [8], TAROT [18], TSS [31], and Distillation [52]. Baselines, implementation, and evaluation details are provided in Appendix B.

5.2 Data Curation for Billion-Parameter Multitask VLAs

5.2.1 RoboTwin 2.0 Simulation Benchmark

Following the official RoboTwin 2.0 protocol, we curate and fine-tune on the demo_clean split, which contains 2,500 demonstrations across 50 tasks totaling 9.34 hours at 16.67 Hz, and evaluate under clean and randomized generalization settings.

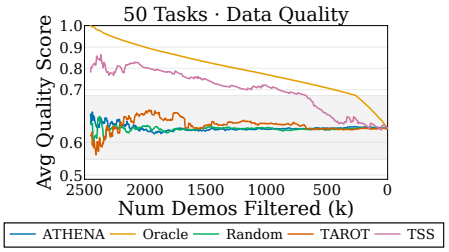


Figure 2: Quality scores over 50 tasks. Shaded band shows a 3× vertical zoom to reveal score variations.

Figures 2 and 3 compare various methods in terms of 50-task quality scores [8, 11] and fine-tuning success rates across different data budgets, respectively. The quality score of the distillation-based method is omitted because it does not perform demonstration-level curation, while its success rate remains low. Although Oracle and TSS retain subsets with higher quality scores, both yield suboptimal success rates. Conversely, despite exhibiting less prominent quality

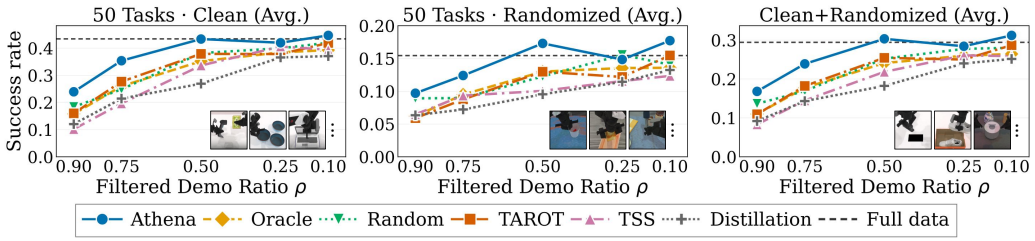


Figure 3: **Average success rate over $K = 50$ tasks.** Results across ρ under clean and randomized evaluation settings; the right panel reports their mean. Dashed lines denote full-data fine-tuning.

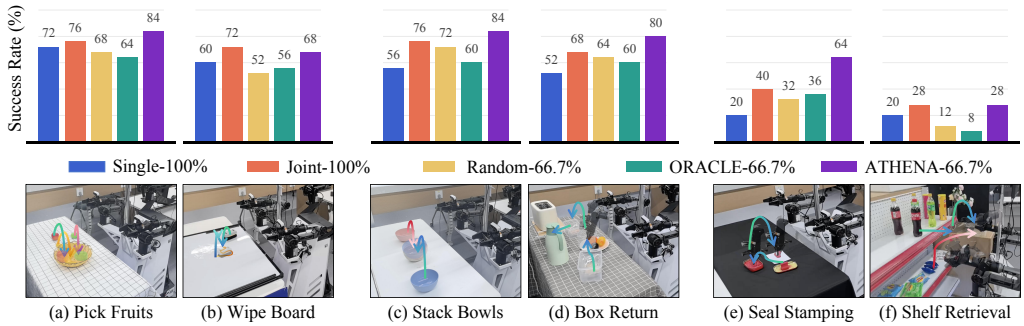


Figure 4: **Real robot evaluation ($K = 6$).** Success rates over 25 trials across six ALOHA tasks of varying difficulty, compared with representative baselines.

scores, ATHENA reaches 44.70% clean and 17.72% randomized success at $\rho = 0.1$, outperforming full-data training (43.42% and 15.44%). Notably, with only 50% of demonstrations retained, ATHENA still matches full-data training under the clean evaluator (43.36% vs. 43.42%) and exceeds it under the randomized evaluator (17.30% vs. 15.44%), yielding a 0.90-point higher average success rate (30.33% vs. 29.43%), corresponding to a cumulative 45.0-point improvement across the 50 tasks. More importantly, even this 50%-data setting remains close to the single-task π_0 fine-tuning performance reported by RoboTwin 2.0 [6] under clean evaluation (43.36% vs. 46.42%) and exceeds it under randomized evaluation (17.30% vs. 16.34%). Unlike single-task training, which requires one fine-tuning and checkpoint per task, ATHENA uses a joint fine-tuning paradigm, reducing computation and storage by tens of times. These contrasting results demonstrate that expert-defined quality is not necessarily predictive of downstream success and highlight the high ROI of ATHENA in multitask VLA fine-tuning.

Furthermore, TAROT achieves moderate gains using Whitened Feature Distance for multitask balancing, but its dependence on precise target-set construction limits further improvements. The Random method also brings improvements, but its uniform retention strategy relies heavily on the quality distribution of the original dataset. Per-task success rates are provided in Appendix C.3. Overall, ATHENA demonstrates superior efficacy in curating data for VLA fine-tuning in simulation.

5.2.2 Real Robot Experiments

To further evaluate ATHENA, we conduct experiments across six real-robot ALOHA tasks spanning three difficulty levels, comprising 720 high-quality demonstrations totaling 6.9 hours collected at 25 Hz. The suite comprises two simple tasks (Pick Fruits, Wipe Board), two medium tasks (Stack Bowls, Box Return), and two challenging long-horizon tasks (Seal Stamping, Shelf Retrieval). To assess positional generalization, we perform 25 trials per task with randomized object positions. Detailed task setups are deferred to Appendix B.2.

As shown in Figure 4, the Single-100% baseline requires six independent fine-tuning checkpoints (totaling 300K training steps and 240GB of storage), yet yields only 46.7% average success. The Joint-100% baseline consolidates this into a single fine-tuning run, drastically reducing these re-

source requirements while improving average success to 60.0%, underscoring the necessity and high ROI of joint training. However, it still struggles with negative-influence data. Random-66.7% and Oracle-66.7% attempt to filter out these data, but instead drop success rates to 50.0% and 47.3%, respectively. In contrast, ATHENA successfully filters out low-utility data while preserving essential multitask knowledge, achieving the highest average success of 68.0% with only 66.7% of the data, corresponding to cumulative gains of 48.0 points over Joint-100% and 128 points over Single-100% across the six tasks. These results demonstrate ATHENA's superior real-robot performance and ROI.

6 Ablation and Discussion

6.1 Ablation of Computational Efficiency

To validate the contributions of Kronecker-compressed featurization and Random Truncated Approximation (RTA) to scalable influence computation, we perform a computational ablation by removing accelerated components from ATHENA. This ablated variant reduces to a standard projected-gradient influence pipeline, where per-sample gradients are explicitly projected, and the projected Gauss–Newton matrix is inverted densely. This unoptimized pipeline matches the computational form of TRAK-style [14] and CUPID-style [8] influence estimation, and therefore serves as a natural reference for evaluating ATHENA's computational speedup. Table 1 reports the total computation time across varying task scales ($K \in \{5, 10, 25, 50\}$, 30.2K–560.5K demonstration timesteps) on 140 GB GPUs. ATHENA achieves a speedup of up to $313.4\times$ over this unoptimized baseline. A detailed per-component breakdown is provided in Appendix C.1.

Table 1: **Computation time (GPU-hours) under data scaling.** Columns denote task counts (K) and corresponding demonstration timesteps.

Method	$K = 5$ (30.2K steps)	$K = 10$ (66.5K steps)	$K = 25$ (291.9K steps)	$K = 50$ (560.5K steps)
w/o Acceleration	~ 446.2	~ 885.5	~ 3297.4	~ 8054.6
ATHENA	$\sim \mathbf{1.1}$	$\sim \mathbf{2.4}$	$\sim \mathbf{14.0}$	$\sim \mathbf{25.7}$
Speedup	$\sim \mathbf{405.6}\times$	$\sim \mathbf{369.0}\times$	$\sim \mathbf{235.5}\times$	$\sim \mathbf{313.4}\times$

6.2 Multitask Joint Fine-tuning Ablation

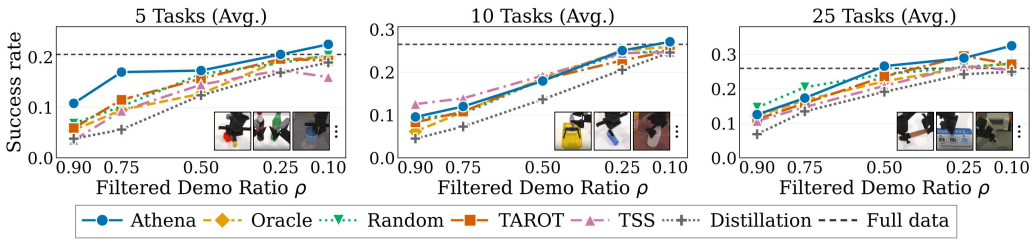


Figure 5: Mean success rates for varying task counts under clean and randomized evaluations.

To ablate ATHENA under varying task counts, we evaluate $K \in \{5, 10, 25\}$ task subsets sampled from the full RoboTwin 50-task benchmark. As shown in Figure 5, at $\rho = 0.1$, ATHENA outperforms other baselines across all three task scales, with the largest improvement at $K = 25$. In this setting, ATHENA achieves 32.48% mean success over both evaluation settings, compared with 25.93% for full-data fine-tuning. Detailed per-setting results are reported in Appendix C.2. At $\rho = 0.5$, the absolute data scale becomes a limiting factor: all methods at $K = 5$ and $K = 10$ remain below full-data fine-tuning, whereas $K = 25$ provides a larger data pool for curation to be effective. This suggests that data curation yields higher returns as data scale increases. Appendix C.4 further analyzes retention imbalance without MII and reports additional single-task curation studies, including real-robot cases on multi-peak actions and spurious correlations.

6.3 Curation Under Mixed Clean-Randomized Data

We next ask whether ATHENA remains effective beyond the clean setting used above. Following the RoboTwin 2.0 clean and randomized (Rand.) configurations, we build a three-task mixed pool and compare it with a compact clean control.

As shown in Figure 6, the 150 demo clean pool yields only limited variation across filtering ratios, since the small-scale dataset has low redundancy and leaves little room for subset optimization. In contrast, scaling to the 1650 demo clean and random pool reveals a clear benefit from curation. At $\rho = 0.50$, ATHENA improves average success from 36.3% to 48.3% under clean evaluation and from 37.0% to 42.3% under randomized evaluation. These results show that ATHENA extracts high ROI subsets from heterogeneous fine-tuning data.

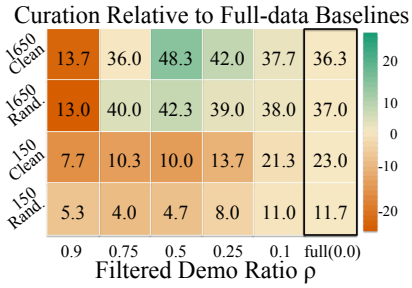


Figure 6: **Curation with mixed data.**

6.4 Cross VLA Model Transferability of Curated Data

Table 2 illustrates that robot demonstration subsets curated via ATHENA on π_0 substantially enhance $\pi_{0.5}$ performance [3, 53], increasing the average success rate across 50 tasks. Both clean and randomized conditions yield average success rates significantly above full-data baselines, indicating that high-influence demonstrations identified by ATHENA generalize across VLA model variants. Detailed per-task success metrics are provided in Appendix C.3.

Table 2: **Cross VLA transfer.** Success rates (%, ↑) for $\pi_{0.5}$ using subsets curated on π_0.

Eval	Filtered Demo Ratio ρ					Full
	0.90	0.75	0.50	0.25	0.10	0.0
Clean	37.86	53.92	63.28	64.16	67.30	57.00
Rand.	16.80	26.74	34.23	37.16	37.77	25.68
AVG	27.33	40.33	48.76	50.66	52.54	41.34

7 Conclusion

In this work, we present ATHENA, an influence function framework tailored for multitask VLA data curation at the billion-parameter scale. To scale influence estimation to VLA models, ATHENA incorporates Kronecker-structured linear-layer gradients and a rank-r Random Truncated Approximation. Furthermore, it formulates Multitask Influence Interaction to balance multitask data curation and improve ROI. Experiments on the 50-task RoboTwin 2.0 benchmark and six real-robot tasks show that ATHENA outperforms full-data fine-tuning and competitive baselines while using fewer demonstrations, indicating its effectiveness for data curation in VLA fine-tuning.

8 Limitations

Pretraining data curation. ATHENA focuses on VLA fine-tuning, while model performance also depends on pretraining with more heterogeneous data across tasks, embodiments, and distributions [3, 54, 55, 56]. Influence estimates from fine-tuning may not directly transfer to this setting, and extending ATHENA to robot VLA pretraining remains future work.

Real-robot rollout cost. Influence-based robot curation currently requires evaluation rollouts to estimate downstream effects [12, 8, 7]. However, collecting rollouts on physical robots for each task incurs substantially higher cost than in simulation. Scaling to broader real-world settings requires more efficient rollout protocols or lower-cost proxies for downstream robot performance.

References

[1] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.

[2] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model. In P. Agrawal, O. Kroemer, and W. Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 2679–2713. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

[3] 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.

[4] F. Lin, Y. Hu, P. Sheng, C. Wen, J. You, and Y. Gao. Data scaling laws in imitation learning for robotic manipulation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=pISLZG7ktL.

[5] S. Belkhale, Y. Cui, and D. Sadigh. Data quality in imitation learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 80375–80395. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/fe692980c5d9732cf153ce27947653a7-Paper-Conference.pdf.

[6] T. Chen, Z. Chen, B. Chen, Z. Cai, Y. Liu, Q. Liang, Z. Li, X. Lin, Y. Ge, Z. Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

[7] H. Lee, T. Min, J. Kim, S. Kang, F. Liu, L. Pinto, and K. Lee. Quality over quantity: Demonstration curation via influence functions for data-centric robot learning. *arXiv preprint arXiv:2603.09056*, 2026.

[8] C. Agia, R. Sinha, J. Yang, R. Antonova, M. Pavone, H. Nishimura, M. Itkina, and J. Bohg. Cupid: Curating data your robot loves with influence functions. *arXiv preprint arXiv:2506.19121*, 2025.

[9] A. Ghorbani and J. Zou. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, pages 2242–2251. PMLR, 2019.

[10] J. Hejna, S. Mirchandani, A. Balakrishna, A. Xie, A. Wahid, J. Tompson, P. Sanketi, D. Shah, C. Devin, and D. Sadigh. Robot data curation with mutual information estimators, 2025. URL https://arxiv.org/abs/2502.08623.

[11] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In A. Faust, D. Hsu, and G. Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 1678–1690. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/mandlekar22a.html.

[12] P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning (ICML)*, 2017.

[13] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. C. Burchfiel, and S. Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi:10.15607/RSS.2023.XIX.026.

[14] S. M. Park, K. Georgiev, A. Ilyas, G. Leclerc, and A. Madry. TRAK: Attributing model behavior at scale. In *International Conference on Machine Learning (ICML)*, 2023.

[15] S. Choe, H. Ahn, J. Bae, K. Zhao, Y. Chung, A. Pratapa, W. Neiswanger, E. Strubell, T. Mitamura, J. Schneider, E. Hovy, R. Grosse, and E. Xing. What is your data worth to gpt? llm-scale data valuation with influence functions. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, pages 145944–145985. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/d6d26053b977f8c589669fd201615119-Paper-Conference.pdf.

[16] B. Mlodozeniec, R. Eschenhagen, J. Bae, A. Immer, D. Krueger, and R. E. Turner. Influence functions for scalable data attribution in diffusion models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 51728–51764, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/804dbf8d3b8eee1ef875c6857efc64eb-Paper-Conference.pdf.

[17] P. W. Koh, K.-S. Ang, H. Teo, and P. S. Liang. On the accuracy of influence functions for measuring group effects. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/a78482ce76496fcf49085f2190e675b4-Paper.pdf.

[18] L. Feng, F. Nie, Y. Liu, and A. Alahi. TAROT: Targeted data selection via optimal transport. In A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 16837–16852. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/feng25l.html.

[19] Y. Tu, Z. Liu, J. W. Ma, and W. Tang. Measuring fine-grained relatedness in multitask learning via data attribution. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=zIDGm96xwg.

[20] S. Dass, A. Khaddaj, L. Engstrom, A. Madry, A. Ilyas, and R. Martín-Martín. DataMIL: Selecting data for robot imitation learning with datamodels. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=AcTsKglDdh.

[21] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, and J. Lee. Transporter networks: Rearranging the visual world for robotic manipulation. In J. Kober, F. Ramos, and C. Tomlin, editors, *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pages 726–747. PMLR, 16–18 Nov 2021. URL https://proceedings.mlr.press/v155/zeng21a.html.

[22] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson. Implicit behavioral cloning. In A. Faust, D. Hsu, and G. Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 158–168. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/florence22a.html.

[23] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In A. Faust, D. Hsu, and G. Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 991–1002. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/jang22a.html.

[24] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[25] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903, 2024. doi:10.1109/ICRA57147.2024.10611477.

[26] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi:10.15607/RSS.2023.XIX.016.

[27] Y. Fan, S. Bai, X. Tong, P. Ding, Y. Zhu, H. Lu, F. Dai, W. Zhao, Y. Liu, S. Huang, Z. Fan, B. Chen, and D. Wang. Long-vla: Unleashing long-horizon capability of vision language action model for robot manipulation. In J. Lim, S. Song, and H.-W. Park, editors, *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 2018–2037. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/fan25a.html.

[28] J. Wen, Y. Zhu, J. Li, Z. Tang, C. Shen, and F. Feng. Dexvla: Vision-language model with plug-in diffusion expert for general robot control. In J. Lim, S. Song, and H.-W. Park, editors, *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 3094–3114. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/wen25b.html.

[29] S. Kuhar, S. Cheng, S. Chopra, M. Bronars, and D. Xu. Learning to discern: Imitating heterogeneous human demonstrations with preference and representation learning. In J. Tan, M. Toussaint, and K. Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 1437–1449. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/kuhar23a.html.

[30] K. Gandhi, S. Karamcheti, M. Liao, and D. Sadigh. Eliciting compatible demonstrations for multi-human imitation learning. In K. Liu, D. Kulic, and J. Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 1981–1991. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/gandhi23a.html.

[31] K. Yang and T. Yang. Less is more: On data redundancy in VLA training. ICLR 2026 Workshop on Data Foundations for Embodied Foundation Models (DATA-FM), 2026. URL https://openreview.net/forum?id=bG8gPlSYvg. OpenReview forum id bG8gPlSYvg. Not arXiv:2601.17815 (a different paper with a similar title).

[32] B. Yu, S. Lian, X. Lin, Z. Shen, Y. Wei, C. Wu, H. Yuan, H. Liu, B. Wang, C. Huang, and K. Chen. Frameskip: Learning from fewer but more informative frames in vla training, 2026. URL https://arxiv.org/abs/2605.13757.

[33] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In J. Tan, M. Toussaint, and K. Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 1820–1864. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/mandlekar23a.html.

[34] C. R. Garrett, A. Mandlekar, B. Wen, and D. Fox. Skillmimicgen: Automated demonstration generation for efficient skill learning and deployment. In P. Agrawal, O. Kroemer, and W. Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 2750–2790. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/garrett25a.html.

[35] T. Yu, T. Xiao, J. Tompson, A. Stone, S. Wang, A. Brohan, J. Singh, C. Tan, D. M, J. Peralta, K. Hausman, B. Ichter, and F. Xia. Scaling Robot Learning with Semantically Imagined Experience. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi:10.15607/RSS.2023.XIX.027.

[36] R. Jia, D. Dao, B. Wang, F. A. Hubis, N. M. Gurel, B. Li, C. Zhang, C. J. Spanos, and D. Song. Efficient task-specific data valuation for nearest neighbor algorithms, 2020. URL https://arxiv.org/abs/1908.08619.

[37] T. Wang, J. Zhu, A. Torralba, and A. A. Efros. Dataset distillation. *CoRR*, abs/1811.10959, 2018. URL http://arxiv.org/abs/1811.10959.

[38] B. Zhao, K. R. Mopuri, and H. Bilen. Dataset condensation with gradient matching. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=mSAKhLYLSsl.

[39] M. Du, S. Nair, D. Sadigh, and C. Finn. Behavior Retrieval: Few-Shot Imitation Learning by Querying Unlabeled Datasets. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi:10.15607/RSS.2023.XIX.011.

[40] S. Nasiriany, T. Gao, A. Mandlekar, and Y. Zhu. Learning and retrieval from prior data for skill-based imitation learning. In K. Liu, D. Kulic, and J. Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 2181–2204. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/nasiriany23a.html.

[41] L.-H. Lin, Y. Cui, A. Xie, T. Hua, and D. Sadigh. Flowretrieval: Flow-guided data retrieval for few-shot imitation learning. In *Conference on Robot Learning*, 2024.

[42] M. Memmel, J. Berg, B. Chen, A. Gupta, and J. Francis. STRAP: Robot sub-trajectory retrieval for augmented policy learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4VHiptx7xe.

[43] J. Hejna, C. A. Bhateja, Y. Jiang, K. Pertsch, and D. Sadigh. Remix: Optimizing data mixtures for large scale imitation learning. In P. Agrawal, O. Kroemer, and W. Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 145–164. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/hejna25a.html.

[44] Q. Sima, W. Xue, and Y. Guo. Promix: Learning optimal data mixtures for robotic imitation via proxy-reference distillation. In *ICRA 2026 Workshop: From Data to Decisions: VLA Pipelines for Real Robots*, 2026. URL https://openreview.net/forum?id=Ee2sKBWzVh.

[45] W. Li, J. Li, P. Zeng, C. S. de Witt, A. Prabhu, and A. Sanyal. Delta-influence: Unlearning poisons via influence functions, 2025. URL https://arxiv.org/abs/2411.13731.

[46] Y. Zhang and M. M. Amiri. Toward efficient influence function: Dropout as a compression tool. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=rapeA5Ha3C.

[47] Z. Tu, C. Chen, and Y. Du. RRInf: Efficient influence function estimation via ridge regression for large language models and text-to-image diffusion models. In C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18494–18507, Suzhou, China, Nov. 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi:10.18653/v1/2025.emnlp-main.933. URL https://aclanthology.org/2025.emnlp-main.933/.

[48] S.-Y. Wang, A. Hertzmann, A. A. Efros, R. Zhang, and J.-Y. Zhu. Fast data attribution for text-to-image models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=3ln8F2nOuA.

[49] Z. Li, W. Zhao, Y. Li, and J. Sun. Where did it go wrong? attributing undesirable llm behaviors via representation gradient tracing. *arXiv preprint arXiv:2510.02334*, 2025.

[50] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[51] Y. Nakatsukasa. Fast and stable randomized low-rank matrix approximation. *arXiv preprint arXiv:2009.11392*, 2020.

[52] K. Chen, Y. Long, S. Li, and M. Shang. Ft-ncfm: An influence-aware data distillation framework for efficient vla models, 2025. URL https://arxiv.org/abs/2511.16233.

[53] K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. R. Equi, C. Finn, N. Fusai, M. Y. Galliker, D. Ghosh, L. Groom, K. Hausman, brian ichter, S. Jakubczak, T. Jones, L. Ke, D. LeBlanc, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, A. Z. Ren, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, J. Tanner, Q. Vuong, H. Walke, A. Walling, H. Wang, L. Yu, and U. Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *9th Annual Conference on Robot Learning*, 2025. URL https://openreview.net/forum?id=vlhoswksBO.

[54] W. Wu, F. Lu, Y. Wang, S. Yang, S. Liu, F. Wang, Q. Zhu, H. Sun, Y. Wang, S. Ma, et al. A pragmatic vla foundation model. *arXiv preprint arXiv:2601.18692*, 2026.

[55] L. Li, Q. Zhang, Y. Luo, S. Yang, R. Wang, F. Han, M. Yu, Z. Gao, N. Xue, X. Zhu, Y. Shen, and Y. Xu. Causal world modeling for robot control, 2026. URL https://arxiv.org/abs/2601.21998.

[56] Y. Wang, X. Li, W. Wang, J. Zhang, Y. Li, Y. Chen, X. Wang, and Z. Zhang. Unified vision-language-action model. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=PklMD8PwUy.

Appendix

A Additional Implementation Details

A.1 Action-level Influence for Flow-matching VLAs

Section 3.1 introduces $\Psi_{a\text{-inf}}$ in Eq. (2) as the action-level term for closed-loop performance influence. Referring to the closed-loop attribution principle of CUPID [8], demonstration-level performance influence is decomposed into action-level influences evaluated on state-action pairs from closed-loop rollouts. In our setting, Eq. (2) retains this decomposition, while the action-level influence is derived for the flow-matching action parameterization of π_0.

Flow-matching action objective. Given a state-action pair $z = (s, a)$, a flow-matching VLA perturbs the action along the linear path

$$x_t = t\epsilon + (1-t)a, \qquad \epsilon \sim \mathcal{N}(0, I), \quad t \in [0, 1], \tag{11}$$

where $x_0 = a$ and $x_1 = \epsilon$. The policy is trained to predict the velocity field along this path. Since

$$\frac{dx_t}{dt} = \epsilon - a, \tag{12}$$

the supervised flow-matching loss is

$$loss_{\text{flow}}(s, a; \theta) = \mathbb{E}_{t,\epsilon}\left[\left\|v_\theta(x_t, s, t) - (\epsilon - a)\right\|^2\right]. \tag{13}$$

This objective defines the fine-tuning loss of the flow-matching policy.

From action likelihood to a flow surrogate. The original action influence in CUPID measures how a training state-action pair (s, a) affects the likelihood of a rollout action (s', a') under the learned policy. In likelihood-based policies, this can be written as an influence on $\log \pi_\theta(a'|s')$. However, for diffusion or flow-matching policies, directly evaluating and differentiating $\log \pi_\theta(a'|s')$ is nontrivial because actions are generated through an iterative generative process rather than a closed-form density. We therefore define a scalar flow-matching surrogate as the rollout-side test quantity $f(\hat{z}; \theta)$ in Eq. (1).

For flow-matching VLAs, we define the square-flow attribution surrogate as

$$f_{\text{sf}}(s, a; \theta) = \mathbb{E}_{t,\epsilon}\left[\left\|v_\theta(x_t, s, t)\right\|^2\right]. \tag{14}$$

Here $x_t = t\epsilon + (1-t)a$, with $\epsilon \sim \mathcal{N}(0, I)$ and $t \in [0, 1]$. This quantity measures the magnitude of the model's velocity response around the queried state-action pair. Unlike the supervised flow-matching loss in Eq. (13), it does not require an expert velocity label on rollout samples and maps the high-dimensional action velocity prediction to a scalar attribution target.

Action-level influence. Let $z = (s, a)$ be a training state-action pair and $\hat{z} = (\hat{s}, \hat{a})$ be a rollout state-action pair. Following the first-order influence form in Eq. (1), we derive the flow-matching action influence as

$$\Psi_{a\text{-inf}}(\hat{z}, z) = -\nabla_\theta f_{\text{sf}}(\hat{z}; \theta^\star)^\top H_\theta^{-1} \nabla_\theta \mathcal{L}(z; \theta^\star), \tag{15}$$

where $\theta^\star$ denotes the fine-tuned VLA model parameters, L is the imitation loss in Section 3.1, and $H_\theta = \nabla_\theta^2 \mathcal{L}(z_t^i; \theta^\star)$ follows the notation of Eq. (1). For the flow-matching VLA model π_0, $\mathcal{L}(z; \theta^\star)$ is given by the supervised flow-matching objective in Eq. (13). In practice, we do not form H_θ^{-1} explicitly. ATHENA computes this interaction in the projected gradient space using the Kronecker-compressed features and the rank-r Random Truncated Approximation described in Section 4 and Appendix A.2.

A.2 Complexity Analysis

Unaccelerated influence estimation complexity. We first derive the time complexity of the main computational bottlenecks in Sections 3 and 4. For multiplying $A \in \mathbb{R}^{a\times b}$ and $B \in \mathbb{R}^{b\times c}$ using standard matrix multiplication, the time complexity is $\mathcal{O}(abc)$. Thus, the projected-gradient operation in Eq. (3), $\Omega^\top g_i$ with $\Omega^\top \in \mathbb{R}^{P\times D}$ and $g_i \in \mathbb{R}^D$, requires $\mathcal{O}(P \cdot D \cdot 1) = \mathcal{O}(DP)$ time per timestep, and $\mathcal{O}(NDP)$ time over N timesteps.

Similarly, forming $G^\top G$ in Eq. (4) multiplies $G^\top \in \mathbb{R}^{P\times N}$ by $G \in \mathbb{R}^{N\times P}$, requiring $\mathcal{O}(P \cdot N \cdot P) = \mathcal{O}(NP^2)$ operations. The result is a $P \times P$ matrix, whose inversion requires $\mathcal{O}(P^3)$ operations. Therefore, the dense projected inverse-Hessian stage has computational complexity $\mathcal{O}(NP^2 + P^3)$. For the experimental setting used in this paper, the π_0 model has $D = 3.3$B parameters, with projected dimension $P = 4096$ and up to $N = 560.5$K demonstration timesteps. At this scale, the unaccelerated projected influence pipeline requires approximately 8054.6 GPU-hours at $K = 50$, as reported in Table 1, highlighting the computational burden of the original $\mathcal{O}(NDP)$ and $\mathcal{O}(NP^2 + P^3)$ pipeline.

Kronecker compressed gradient featurization. We expand the Kronecker-structured projection used in Eq. (5). For a linear layer ℓ such as an MLP layer, let $W^\ell \in \mathbb{R}^{d_\ell^{\text{out}}\times d_\ell^{\text{in}}}$, with layer parameter dimension $D_\ell = d_\ell^{\text{out}} d_\ell^{\text{in}}$. For timestep i, the input activation and backpropagated error are $x_i^\ell \in \mathbb{R}^{d_\ell^{\text{in}}}$ and $\delta_i^\ell \in \mathbb{R}^{d_\ell^{\text{out}}}$, respectively. The per-timestep weight gradient is

$$G_i^\ell = \nabla_{W^\ell} L(z_i; \theta^\star) = \delta_i^\ell (x_i^\ell)^\top. \tag{16}$$

After vectorization, using $\text{vec}(ab^\top) = b \otimes a$, where $\otimes$ denotes the Kronecker product, we obtain $g_i^\ell = \text{vec}(G_i^\ell) = x_i^\ell \otimes \delta_i^\ell$.

A dense projection applies $\Omega_\ell^\top g_i^\ell$ with $\Omega_\ell \in \mathbb{R}^{D_\ell \times P}$, where P is the projected gradient dimension in Eq. (3), requiring $\mathcal{O}(D_\ell P)$ operations for layer ℓ. Following LoGRA [15], ATHENA imposes a Kronecker-product structure on the projection matrix:

$$\Omega_\ell = P_{\text{in}}^\ell \otimes P_{\text{out}}^\ell. \tag{17}$$

By the mixed-product property of Kronecker products,

$$\Omega_\ell^\top g_i^\ell = \big((P_{\text{in}}^\ell)^\top x_i^\ell\big) \otimes \big((P_{\text{out}}^\ell)^\top \delta_i^\ell\big). \tag{18}$$

Equivalently, the same projected feature can be computed in matrix form as

$$\bar{g}_i^\ell = (P_{\text{out}}^\ell)^\top \delta_i^\ell (x_i^\ell)^\top P_{\text{in}}^\ell, \tag{19}$$

which is the compact bilateral projection in Eq. (5).

This bilateral form avoids materializing the full layer gradient g_i^ℓ and performs projection directly in activation space. When $d_\ell^{\text{in}} \approx d_\ell^{\text{out}} \approx \sqrt{D_\ell}$ and the two projected activation dimensions are both on the order of $\sqrt{P}$, the per-layer projection cost reduces from $\mathcal{O}(D_\ell P)$ to $\mathcal{O}(\sqrt{D_\ell P})$ [15]. Applying this reduction across projected layers and N timesteps give the overall projected-gradient construction cost of $\mathcal{O}(N\sqrt{DP})$, compared with the dense $\mathcal{O}(NDP)$ cost.

Random Truncated Approximation. After Kronecker compressed featurization, ATHENA stacks the projected timestep features into $G \in \mathbb{R}^{N\times P}$, where each row is $\phi_i^\top$. Following Eq. (4), the dense projected inverse requires forming and inverting $G^\top G + \lambda I_P$, with complexity $\mathcal{O}(NP^2 + P^3)$.

To avoid this dense projected inverse, ATHENA applies a rank-r Random Truncated Approximation (RTA) [50, 51] to G. Specifically, we approximate $G \approx U_r \Sigma_r V_r^\top$, where $V_r \in \mathbb{R}^{P\times r}$ and $r \ll P$, so that $G^\top G \approx V_r \Sigma_r^2 V_r^\top$. Each projected feature is then mapped to the retained subspace as $\tilde{\phi}_i = V_r^\top \phi_i$, leading to the rank-$r$ influence surrogate in Eq. (6):

$$\widehat{\psi}_{\text{RTA}}(z_{\text{te}}, z_{\text{tr}}) = \tilde{\phi}_{\text{te}}^\top (\Sigma_r^2 + \lambda I_r)^{-1} \tilde{\phi}_{\text{tr}}.$$

The randomized range-finding step and the projection of all N timestep features into the retained subspace both scale as $\mathcal{O}(NPr)$ up to lower-order terms. Therefore, RTA reduces the projected inverse-Hessian stage from the dense $\mathcal{O}(NP^2 + P^3)$ computation to $\mathcal{O}(NPr)$.

Algorithm 1 Multitask Influence Interaction (MII) Score Computation

Input: Demonstration set $\mathcal{D} = \{i\}_{i=1}^{n}$, task index $c(i)$, rollout set $\mathcal{E}$, reward function $R(\cdot)$, influence score $S_{j,i}$

Partition rollouts into task-specific sets $\{\mathcal{E}_k\}_{k=1}^{K}$

for each demonstration $i \in \mathcal{D}$ **do**

Compute task-local influence:

$$\widetilde{\Psi}_{\pi\text{-inf}}^{c(i)}(i) = \frac{1}{H_i} \sum_{\tau_j \in \mathcal{E}_{c(i)}} R(\tau_j) S_{j,i}$$

Compute cross-task influence:

$$\widetilde{\Psi}_{\pi\text{-inf}}^{\text{all}-c(i)}(i) = \frac{1}{H_i} \sum_{\tau_j \in \mathcal{E} \setminus \mathcal{E}_{c(i)}} R(\tau_j) S_{j,i}$$

end for

Compute ranks within each task group:

$$r_i^k = \text{rank}_{\downarrow}^{k}(\cdot)$$

Normalize utilities:

$$u_i^{c(i)},\ u_i^{\text{all}-c(i)}$$

Compute final score:

$$f_i^{\text{MII}} = u_i^{c(i)} \cdot u_i^{\text{all}-c(i)}$$

Output: ranked demonstrations by f_i^{MII}

A.3 Multitask data curation

Algorithm 1 summarizes the computation of the Multitask Influence Interaction (MII) score introduced in Section 4. For each demonstration, MII separately estimates its task-local utility from its own task and its cross-task utility over rollouts from all other tasks. The two utilities are then rank-normalized within each task group and combined to obtain the final curation score. When $K = 1$, no other tasks exist, and the procedure reduces to standard single-task influence-based data curation. Single-task simulation and real-robot results are provided in Appendix C.4 and Figs. 9, 10, and 11.

B Experimental Details

This section collects extended experimental specifications referenced from Section 5.

B.1 RoboTwin 2.0 Simulation Benchmark

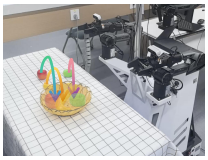
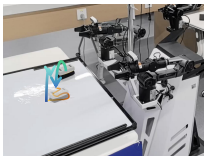
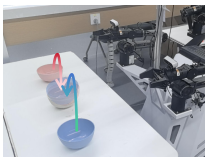
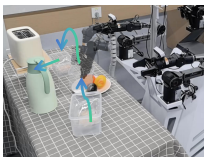
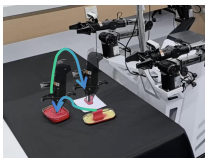
We conduct simulation experiments on RoboTwin 2.0 using its official simulator, task definitions, success criteria, and rollout evaluation protocol. We use the K=50 task suite to evaluate large-scale multitask VLA fine-tuning and data curation. We report results under two RoboTwin 2.0 evaluation configurations. The demo_clean configuration corresponds to the Easy setting, while demo_randomized corresponds to the Hard setting with stronger scene randomization and more challenging initial conditions [6].

B.2 Real-Robot Task Setups

We summarize the six real-robot tasks in Table 3, including task names, execution illustrations, and policy prompts.

All real-robot experiments are conducted on an AgileX Cobot Magic platform, a Mobile ALOHA-style bimanual robot with two forward-reaching arms. The robot uses three RGB cameras: one

Table 3: Descriptions of the six real-robot manipulation tasks, each with 120 demonstrations.

Task Setup	Description	Task Setup	Description
	**Pick Fruits:** Pick the fruits and put them in the basket.		**Wipe Board:** Pick up the eraser and wipe the whiteboard.
	**Stack Bowls:** Stack the bowls on the table, with the middle one.		**Box Return:** Move the box under the shelf.
	**Seal Stamping:** Dip the stamp in ink and stamp.		**Shelf Retrieval:** Use the left arm to retrieve a Coke from the upper shelf and transfer it to the right arm for placement into the basket, then use the right arm to retrieve Oreos from the lower shelf and place them into the basket.

upper-body camera and two wrist cameras mounted near the left and right end-effectors. During data collection, we record synchronized RGB observations from the three cameras and teleoperated expert trajectories. We fully fine-tune the π_0, a VLA model, using the OpenPI JAX framework on the collected real-robot demonstrations, with a batch size of 32 for 50k steps. During deployment, an onboard NVIDIA RTX 4090 GPU is used for real-time action inference and execution.

B.3 Baselines

We compare ATHENA to several data curation baselines, including Oracle, Random, TAROT, TSS, and Distillation. For all baselines, we use the same filtering ratios $\rho \in \{0.10, 0.25, 0.50, 0.75, 0.90\}$. All curated subsets are fine-tuned and evaluated with the same settings as ATHENA.

Oracle. Following the Oracle definitions in RoboMimic and CUPID [11, 8], demonstration length is used as a proxy for dataset quality. Specifically, Oracle assumes that shorter length demonstrations correspond to more efficient task completion and therefore indicate higher demonstration quality. For each demonstration ξ_i, the Oracle baseline assigns the following quality score:

$$S_{\text{oracle}}(\xi_i) = -T(\xi_i), \tag{20}$$

where $T(\xi_i)$ denotes the demonstration length, i.e., the completion time of ξ_i. Demonstrations are ranked in descending order of S_{oracle}, so demonstrations with shorter length are retained before longer ones. However, Oracle does not accurately model downstream task success, and as shown in Figures 2 and 3 (see also CUPID [8]), the expert-based quality scores do not always align with downstream performance.

Random. Random is a data-agnostic baseline that uniformly samples demonstrations from the training pool under each filtering ratio.

TAROT. We adapt TAROT [18] as a target-distribution matching baseline for multitask demonstration curation. It selects training demonstrations by matching candidate demonstrations to a target set in a whitened gradient-feature space. In our setting, the candidate set consists of training demonstrations, while the target set consists of held-out or evaluation episodes from the corresponding

task suite. TAROT constructs a candidate-target similarity matrix and solves an entropic optimal-transport problem with Sinkhorn iterations. The transport plan prioritizes demonstrations that better cover the target distribution. However, unlike ATHENA, TAROT remains sensitive to target-set quality, limiting its multitask curation performance.

Temporal Surprise Score (TSS). TSS [31] is a temporal-difficulty sampling baseline that values demonstrations by the magnitude of their action changes over time. For RoboTwin 2.0 ALOHA tasks, each action a_t is a 14-dimensional dual-arm command, consisting of two 6-D arm-motion commands and two scalar gripper commands. We denote the arm-motion component as $m_t = [a_t^0, \ldots, a_t^5, a_t^7, \ldots, a_t^{12}]$ and the two gripper commands as $g_t^L = a_t^6$ and $g_t^R = a_t^{13}$. The frame-level TSS score is defined as

$$S(t) = D_{\cos}(m_t, m_{t-1}) + \gamma \left(|g_t^L - g_{t-1}^L| + |g_t^R - g_{t-1}^R| \right), \tag{21}$$

where $D_{\cos}$ is the cosine distance between consecutive arm-motion commands, and γ controls the relative weight of gripper changes. We set $\gamma = 1$ in all experiments. We aggregate frame-level scores into a demonstration-level score and rank demonstrations in descending order, so trajectories with larger temporal action changes are retained first.

Distillation. Distillation is a demonstration-level baseline that selects trajectories based on their diversity with respect to task-level action patterns. For each task, we compute a prototype trajectory from the average resampled demonstrations, and score each demonstration by its deviation from this prototype:

$$S_{\text{distill}}(\xi_i) = \left\| \text{Resample}(\xi_i) - \mu_{\text{task}(i)} \right\|_F, \tag{22}$$

where $\mu_{\text{task}(i)}$ denotes the task-level prototype trajectory, computed by averaging the resampled demonstrations from the same task as ξ_i. Demonstrations with larger scores are treated as less redundant and are retained first, encouraging diversity in the curated subset. However, this score is only a trajectory-space proxy: it does not use policy rollouts, gradient information, or downstream performance feedback, and may therefore select diverse demonstrations that are not necessarily beneficial for improving policy success.

C Additional Experiments and Discussion

C.1 Analysis of Computational Efficiency

We ablate ATHENA's two core acceleration components, Kronecker compressed gradient featurization and the Random Truncated Approximation for the Hessian, to demonstrate their necessity for scaling influence calculation to large-scale, widely adopted VLA models. We conduct this evaluation on a multitask setup consisting of $K = 50$ tasks with $N \approx 560{,}500$ total timesteps, benchmarking the 3.3B-parameter π_0 model as a representative case on a 140 GB GPU platform (Table 4).

As shown, directly applying naive CUPID to π_0 is computationally prohibitive. Explicitly materializing the full D-dimensional gradient causes immediate out-of-memory errors even on high-end 140 GB hardware, forcing a strict batch size of 1 and inflating featurization time to 8004.6 GPU·h. ATHENA circumvents this spatial bottleneck via Kronecker compressed projection directly onto layer-local activation buffers, reducing peak memory to $\sim$ 1 GB and accelerating featurization to 23.2 GPU·h under a distributed batched setting. For the Hessian step, dense inversion requires 50 GPU·h, whereas ATHENA's truncation approximation requires only 2.5 GPU·h. Combined, ATHENA reduces the total 50-task attribution overhead from 8054.6 to 25.7 GPU·h, achieving a $313.4\times$ speedup. Crucially, while a cost of nearly 8054.6 GPU·h on a modest 50-task setup fundamentally shuts down any possibility of scaling naive methods to broader multitask regimes, ATHENA's high efficiency unlocks a viable path toward open-ended scaling across hundreds of tasks and millions of timesteps.

Table 4: **Complexity ablation on π_0 for $K{=}50$ tasks.** VRAM is incremental, and GPU·h influences computation time. † marks per-sample full-gradient memory.

	Time complexity	Incremental VRAM	GPU·h
Gradient featurization (50 tasks)			
w/o Kronecker	$\mathcal{O}(KN \cdot DP)$	$\sim$13 GB† (+OOM)	8004.6
ATHENA	$\mathcal{O}(KN \cdot Lkd)$	$\sim$ **1** GB	**23.2**
Hessian approximation			
w/o RTA	$\mathcal{O}(NP^2 + P^3)$	—	50.0
ATHENA (Random Truncation)	$\mathcal{O}(KN \cdot D_{\text{grad}} r)$	—	**2.5**
Total, 50 tasks	—	—	8054.6 **vs** 25.7

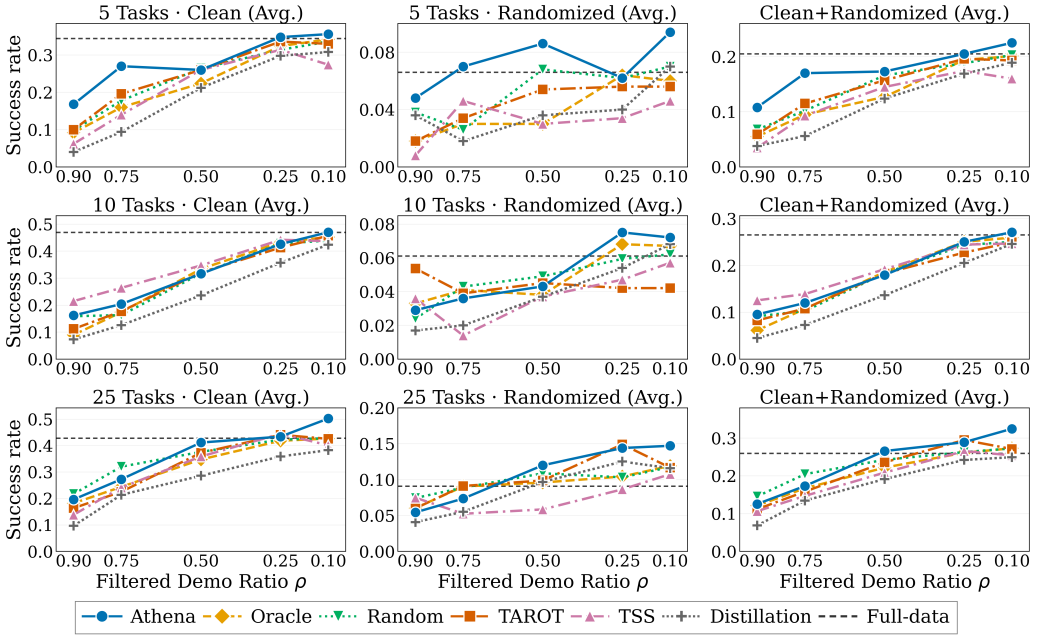


Figure 7: Success rate trends across task counts and filtering ratios ρ. Each row corresponds to $K \in \{5, 10, 25\}$, and columns report clean, randomized, and averaged success rates. Dashed lines denote full-data fine-tuning.

C.2 Per-Setting Multitask Ablation Results

This section complements Section 6.2 by decomposing the averaged multitask results into clean and randomized evaluations. As shown in Figure 7, ATHENA remains competitive across both settings and all task scales, indicating that the gains reported in the main text are not dominated by a single evaluation setting.

At $\rho = 0.1$, ATHENA exceeds full-data performance under both evaluations for all three task scales. For $K = 5$, it achieves 35.60% clean-evaluation success and 9.40% randomized-evaluation success, compared with full-data baselines of 34.40% and 6.60%. For $K = 10$, ATHENA reaches 47.00% and 7.21%, surpassing 46.90% and 6.11%. For $K = 25$, the gains become more pronounced, with 50.24% clean-evaluation success and 14.72% randomized-evaluation success, compared with 42.76% and 9.09%.

At $\rho = 0.5$, the per-setting curves are consistent with the data-scale effect discussed in Section 6.2. Smaller task subsets ($K = 5, 10$) provide a limited candidate set for curation, leading to weaker or unstable gains, whereas $K = 25$ allows ATHENA to remove low-utility demonstrations while

preserving task coverage. These results provide per-setting evidence for the main-text observation that curation yields higher returns as task diversity increases.

C.3 Per-task Success Analysis

Tables 5 and 6 report per-task evaluation results of jointly fine-tuned 50-task policies under different filtering ratios. We refer to Lingbo-VA [55] to stratify the 50 tasks into 30 H1, 15 H2, and 5 H3.

For the detailed breakdown of Figure 3, corresponding to the main $\pi_0 \to \pi_0$ setting (Table 5), at the aggregate level $\rho = 0.5$ preserves clean performance ($43.42\% \to 43.36\%$) while improving randomized success from 15.44% to 17.30%. With lighter filtering ($\rho = 0.1$), both clean and randomized averages increase to 44.70% and 17.72%, respectively. Specifically, per-horizon analysis highlights contributions from different task groups.

For H1 tasks, $\rho = 0.1$ increases the clean average from 47.7% to 50.1% and randomized from 19.5% to 21.8%, with notable gains on contact- or alignment-sensitive tasks, e.g., *place_shoe* ($43.0\% \to 65.0\%$), *click_bell* ($66.0\% \to 83.0\%$), and *press_stapler* ($70.0\% \to 75.0\%$). For H2 tasks, moderate filtering ($\rho = 0.5$) yields the best group averages: clean from 39.1% to 40.8%, randomized from 10.47% to 13.27%, with representative gains on *place_cans_plasticbox* ($55.0\% \to 77.0\%$) and *place_bread_skillet* ($35.0\% \to 49.0\%$). H3 tasks benefit mainly in clean evaluation at $\rho = 0.1$ ($24.8\% \to 28.0\%$) and in randomized at $\rho = 0.5$ ($5.8\% \to 9.6\%$), e.g., *stack_bowls_three* ($61.0\% \to 72.0\%$) and *blocks_ranking_rgb* ($16.0\% \to 20.0\%$). These observations demonstrate that highly redundant tasks benefit from removing low-utility data, whereas precise or long-horizon tasks are more sensitive and may see declines when essential data is removed [8].

Table 6 shows that the same π_0-curated subsets also improve $\pi_{0.5}$ performance (corresponding to Table 2). Using these subsets increases the average to 67.30% clean and 37.77% randomized at $\rho = 0.1$, and even with only half of the data retained ($\rho = 0.5$), it reaches 63.28% clean and 34.23% randomized, both exceeding the full-data baselines (57.00% clean, 25.68% randomized), demonstrating cross-model applicability.

C.4 Retention Balance, Single-Task Curation, and Real-Robot Failure Modes

To further ablate the role of Multitask Influence Interaction (MII), we visualize the retained task distributions after data curation in Fig. 8. We consider the six-task real-robot setting with 120 demonstrations per task and an overall retention ratio of 66.7%. Without MII, naively ranking demonstrations with a single global influence score results in a highly skewed retained set: Pick Fruits, Shelf Retrieval, and Wipe Board retain 115, 113, and 104 demonstrations, respectively, whereas Stack Bowls retains only 13 and is nearly eliminated. This occurs because raw influence magnitudes are not directly comparable across heterogeneous tasks with different horizons, motion patterns, and cross-task coupling; as a result, the naive global ranking can overemphasize dataset-level contribution while ignoring task-local importance. With MII, ATHENA combines task-local and cross-task influence utilities, yielding a balanced retained distribution and preventing task-level collapse under the same retention ratio.

***RoboTwin* single-task analysis.** Fig. 9 reports single-task fine-tuning results on four RoboTwin tasks. Light curation consistently improves policy performance: filtering only 10% of demonstrations outperforms full-data training on all tasks, improving success from 55% to 56% on handover, 35.5% to 47.5% on turn switch, 55% to 82% on place cans plasticbox, and 26% to 39% on move pillbottle pad. This suggests that a fraction of demonstrations can harm policy performance rather than improve it, consistent with prior findings in influence-based robot data curation [8]. With stronger filtering, performance becomes task-dependent: handover is more data-hungry and degrades as more demonstrations are removed, likely because it requires sufficient coverage of diverse interaction and transfer states; in contrast, the other three tasks remain robust to data reduction. At $\rho = 0.5$, ATHENA matches or surpasses full-data training on three out of four tasks.

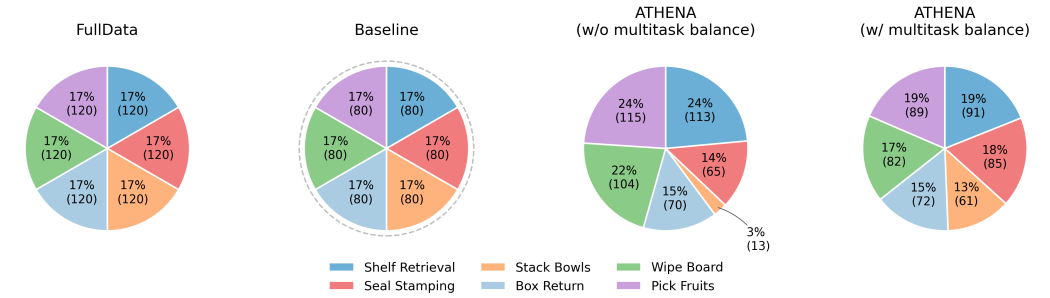


Figure 8: Distributions of the retained 66.7% demonstrations across the six real-robot tasks.

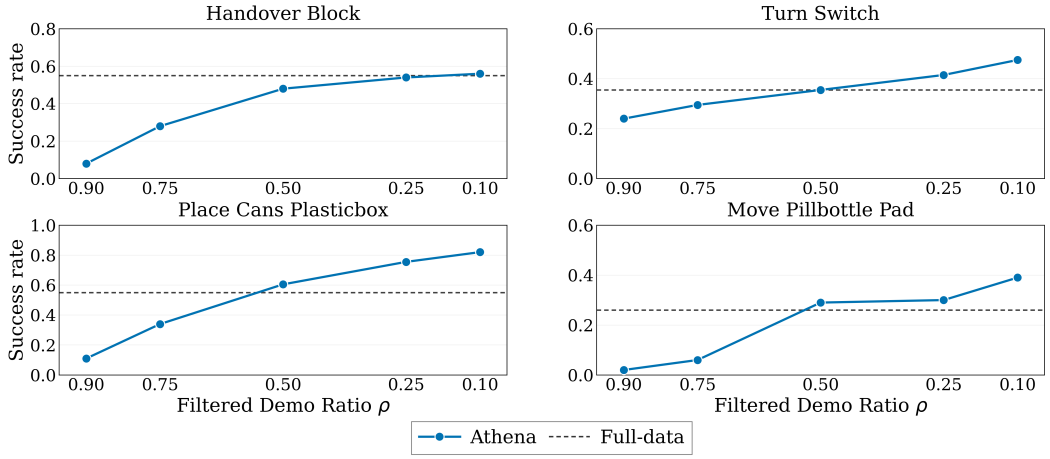


Figure 9: Single-task fine-tuning results on four RoboTwin tasks.

***Multi-peak Action Distribution* task analysis.** As shown in Fig. 10, the Stack Bowls task exhibits a multi-peak action distribution in the collected demonstrations. Specifically, under similar task setups, the demonstrations contain two valid bimanual execution modes: a left-first mode accounting for roughly one-third of the demonstrations, where the left arm initiates the stacking sequence before the right arm, and a right-first mode accounting for roughly two-thirds, where the execution order is reversed. Although both modes can complete the task, their coexistence in the training data introduces ambiguity into the learned action distribution. During inference, the policy may combine action tendencies from both modes, causing the two arms to move toward different candidate bowls simultaneously and leading to execution hesitation or occasional stalling. Applying ATHENA to curate the task-specific training data increases the success rate from 56% to 68%, suggesting that data curation mitigates the effect of multi-peak action modes. However, its success rate remains limited when evaluated across different bowl positions, as the policy lacks compositional generalization over spatial bowl arrangements without multitask training.

***Spurious Association* task analysis.** As shown in Fig. 11, the Box Return task exhibits a spurious association between scene appearance and action mode in the collected demonstrations. The training data contains three subgroups with an approximate ratio of 3:2:1. In the largest subgroup, the scene uses a dark tablecloth with distractor objects and no obstacle, and the demonstrated behavior follows a dragging mode. In the second subgroup, the scene uses a light tablecloth with no distractor objects but includes an obstacle, and the demonstrated behavior follows a transfer mode. The smallest subgroup contains both distractor objects and an obstacle, while still requiring the transfer mode. Due to the imbalance among these subgroups, the policy can incorrectly associate background appearance or distractor objects with the dragging behavior, rather than relying on the obstacle configuration to determine the appropriate action mode. At test time, this spurious association may cause the policy to predict dragging even when a transfer is required, leading to a collision with the obstacle. By

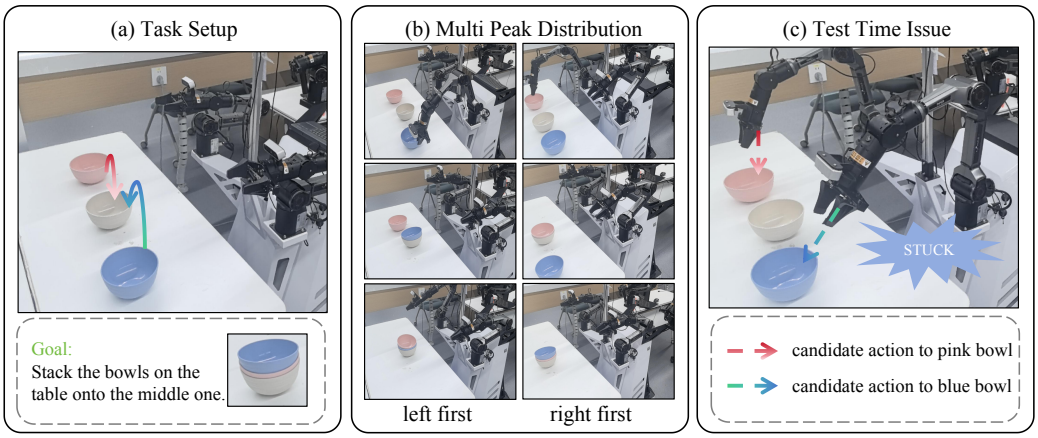


Figure 10: **Illustration of the multi-peak action distribution in the Stack Bowls task.** (a) The task requires stacking the bowls onto the middle bowl. (b) The collected demonstrations contain two valid bimanual execution modes, corresponding to left-first and right-first manipulation orders. (c) At test time, the learned policy may mix the two modes and command the two arms toward different candidate bowls simultaneously, leading to execution hesitation or a stuck state.

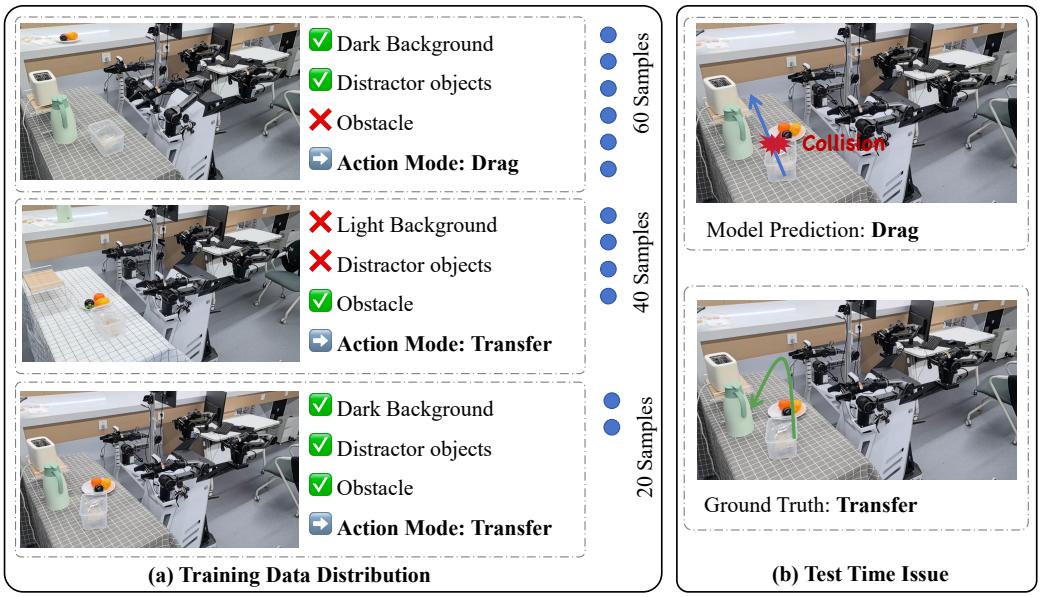


Figure 11: **Illustration of spurious associations in the Box Return task.** (a) The collected demonstrations contain imbalanced subgroups that couple visual appearance, obstacle presence, and action modes. The smallest subgroup provides disambiguating examples where distractor objects and an obstacle coexist, but the correct behavior remains transfer. (b) At test time, the policy may rely on spurious visual cues and predict dragging when transfer is required, leading to obstacle collision.

applying ATHENA to curate the task-specific training data, the relative contribution of the disambiguating subgroup is increased, reducing the effect of spurious visual correlations and improving the selection of the correct action mode. This leads to a substantial improvement in performance, increasing the success rate from 52% to 74%.

Table 5: Per-task success rates under different filtering ratios on the 50-task benchmark. Values are success rates in percentage points. The full-data result is denoted by $\rho = 0.0$.

Task	Horizon	$\rho = 0.9$		$\rho = 0.75$		$\rho = 0.5$		$\rho = 0.25$		$\rho = 0.1$		$\rho = 0.0$	
		Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.
Adjust Bottle	1	35.00	7.00	65.00	3.00	62.00	4.00	41.00	11.00	56.00	9.00	70.00	10.00
Beat Block Hammer	1	17.00	7.00	28.00	12.00	62.00	12.00	62.00	11.00	64.00	13.00	60.00	13.00
Blocks Ranking Rgb	3	0.00	0.00	5.00	0.00	14.00	2.00	45.00	2.00	20.00	1.00	16.00	3.00
Blocks Ranking Size	3	0.00	0.00	3.00	0.00	8.00	1.00	0.00	0.00	9.00	0.00	11.00	1.00
Click Alarmclock	1	78.00	33.00	82.00	27.00	78.00	41.00	90.00	32.00	86.00	47.00	85.00	19.00
Click Bell	1	91.00	54.00	92.00	39.00	90.00	49.00	83.00	39.00	83.00	45.00	66.00	18.00
Dump Bin Bigbin	1	42.00	30.00	55.00	49.00	74.00	34.00	69.00	37.00	63.00	30.00	75.00	37.00
Grab Roller	1	64.00	41.00	79.00	48.00	93.00	60.00	100.00	52.00	93.00	75.00	99.00	67.00
Handover Block	2	0.00	0.00	11.00	1.00	15.00	0.00	11.00	0.00	21.00	0.00	5.00	0.00
Handover Mic	2	61.00	1.00	57.00	2.00	29.00	2.00	51.00	2.00	35.00	6.00	49.00	4.00
Hanging Mug	2	2.00	0.00	10.00	3.00	10.00	1.00	6.00	2.00	15.00	2.00	9.00	0.00
Lift Pot	1	8.00	0.00	16.00	0.00	23.00	0.00	26.00	0.00	29.00	0.00	20.00	0.00
Move Can Pot	1	5.00	0.00	12.00	0.00	48.00	7.00	40.00	2.00	10.00	2.00	19.00	1.00
Move Pillbottle Pad	1	4.00	0.00	14.00	1.00	19.00	3.00	21.00	2.00	24.00	7.00	30.00	5.00
Move Playingcard Away	1	20.00	1.00	61.00	19.00	71.00	32.00	59.00	15.00	71.00	17.00	56.00	30.00
Move Stapler Pad	1	0.00	0.00	2.00	0.00	1.00	1.00	1.00	0.00	1.00	0.00	1.00	0.00
Open Laptop	1	55.00	23.00	78.00	19.00	67.00	20.00	66.00	12.00	58.00	13.00	49.00	21.00
Open Microwave	1	14.00	24.00	34.00	29.00	57.00	16.00	53.00	25.00	41.00	45.00	60.00	39.00
Pick Diverse Bottles	2	6.00	1.01	48.00	7.07	50.00	3.03	47.00	1.01	59.00	5.05	48.00	2.02
Pick Dual Bottles	2	5.00	1.00	55.00	10.00	67.00	12.00	56.00	10.00	62.00	18.00	79.00	7.00
Place A2b Left	1	15.00	1.00	19.00	2.00	23.00	11.00	24.00	9.00	19.00	8.00	24.00	7.00
Place A2b Right	1	8.00	3.00	8.00	4.00	19.00	4.00	25.00	4.00	27.00	6.00	31.00	13.00
Place Bread Basket	1	4.00	1.00	14.00	4.00	31.00	20.00	30.00	27.00	37.00	18.00	38.00	31.00
Place Bread Skillet	2	1.00	0.00	17.00	2.00	49.00	11.00	61.00	13.00	33.00	17.00	35.00	17.00
Place Burger Fries	2	9.00	4.00	5.00	1.00	39.00	58.00	11.00	32.00	9.00	34.00	17.00	36.00
Place Can Basket	2	18.00	0.00	28.00	1.00	51.00	8.00	27.00	4.00	47.00	2.00	46.00	1.00
Place Cans Plasticbox	2	18.00	6.00	17.00	11.00	77.00	16.00	16.00	28.00	51.00	28.00	55.00	37.00
Place Container Plate	1	69.00	43.00	72.00	44.00	35.00	36.00	40.00	23.00	64.00	51.00	52.00	23.00
Place Dual Shoes	2	4.00	6.00	18.00	10.00	42.00	12.00	34.00	14.00	41.00	31.00	43.00	19.00
Place Empty Cup	1	24.00	7.00	27.00	7.00	52.00	18.00	64.00	14.00	81.00	30.00	72.00	35.00
Place Fan	1	2.00	1.00	4.00	2.00	4.00	0.00	7.00	2.00	20.00	5.00	17.00	7.00
Place Mouse Pad	1	4.00	0.00	15.00	4.00	15.00	2.00	28.00	5.00	20.00	1.00	13.00	4.00
Place Object Basket	2	24.00	2.00	58.00	7.00	68.00	22.00	62.00	7.00	71.00	3.00	72.00	3.00
Place Object Scale	1	5.00	0.00	14.00	3.00	16.00	6.00	20.00	3.00	22.00	2.00	9.00	3.00
Place Object Stand	1	35.00	19.00	45.00	16.00	68.00	28.00	47.00	19.00	57.00	27.00	41.00	7.00
Place Phone Stand	1	8.00	0.00	14.00	3.00	25.00	7.00	34.00	2.00	32.00	2.00	35.00	5.00
Place Shoe	1	5.00	3.00	34.00	9.00	45.00	26.00	40.00	22.00	65.00	33.00	43.00	18.00
Press Stapler	1	62.00	15.00	55.00	20.00	57.00	24.00	69.00	22.00	75.00	16.00	70.00	18.00
Put Bottles Dustbin	3	14.00	1.00	35.00	8.00	25.00	6.00	26.00	6.00	37.00	5.00	31.00	8.00
Put Object Cabinet	2	18.00	1.00	35.00	1.00	24.00	3.00	18.00	1.00	17.00	2.00	19.00	2.00
Rotate QRcode	1	39.00	3.00	53.00	12.00	54.00	12.00	57.00	2.00	63.00	5.00	66.00	8.00
Scan Object	2	5.00	1.00	11.00	1.00	11.00	2.00	11.00	4.00	12.00	7.00	13.00	6.00
Shake Bottle	1	87.00	56.00	97.00	54.00	92.00	71.00	92.00	67.00	96.00	65.00	98.00	68.00
Shake Bottle Horizontally	1	89.00	52.00	96.00	53.00	90.00	60.00	93.00	64.00	94.00	56.00	90.00	68.00
Stack Blocks Three	3	0.00	0.00	0.00	0.00	2.00	0.00	2.00	1.00	2.00	0.00	5.00	0.00
Stack Blocks Two	2	14.00	2.00	8.00	2.00	23.00	1.00	27.00	13.00	26.00	3.00	37.00	5.00
Stack Bowls Three	3	28.00	9.00	55.00	20.00	62.00	39.00	66.00	25.00	72.00	33.00	61.00	17.00
Stack Bowls Two	2	61.00	18.00	87.00	33.00	87.00	48.00	90.00	38.00	92.00	35.00	89.00	18.00
Stamp Seal	1	10.00	0.00	10.00	3.00	20.00	1.00	18.00	6.00	22.00	8.00	16.00	2.00
Turn Switch	1	10.00	8.00	11.00	14.00	24.00	13.00	34.00	10.00	31.00	18.00	26.00	9.00
Average	–	**23.94**	**9.70**	**35.38**	**12.40**	**43.36**	**17.30**	**42.00**	**14.84**	**44.70**	**17.72**	**43.42**	**15.44**

Table 6: Per-task success rates when using π_0-selected data to train $\pi_{0.5}$. Values are success rates in percentage points. The full-data result is denoted by $\rho = 0.0$.

Task	Horizon	$\rho = 0.9$		$\rho = 0.75$		$\rho = 0.5$		$\rho = 0.25$		$\rho = 0.1$		$\rho = 0.0$	
		Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.
Adjust Bottle	1	61.00	39.00	44.00	48.00	97.00	75.00	100.00	89.00	100.00	87.00	97.00	54.00
Beat Block Hammer	1	56.00	8.00	71.00	7.00	78.00	10.00	90.00	32.00	80.00	14.00	3.00	0.00
Blocks Ranking Rgb	3	6.00	4.00	6.00	1.00	11.00	2.00	10.00	6.00	31.00	3.00	63.00	8.00
Blocks Ranking Size	3	4.00	0.00	11.00	2.00	28.00	15.00	32.00	19.00	22.00	11.00	33.00	11.00
Click Alarmclock	1	78.00	49.00	95.00	65.00	91.00	69.00	83.00	55.00	100.00	67.00	22.00	62.00
Click Bell	1	76.00	65.00	97.00	95.00	99.00	89.00	93.00	76.00	91.00	95.00	6.00	85.00
Dump Bin Bigbin	1	7.00	1.00	87.00	61.00	80.00	77.00	95.00	75.00	85.00	74.00	97.00	30.00
Grab Roller	1	74.00	58.00	91.00	85.00	100.00	91.00	100.00	95.00	98.00	97.00	100.00	63.00
Handover Block	2	6.00	0.00	37.00	0.00	56.00	1.00	52.00	1.00	29.00	1.00	42.00	0.00
Handover Mic	2	63.00	1.00	70.00	1.00	53.00	1.00	43.00	1.00	62.00	0.00	37.00	2.00
Hanging Mug	2	1.00	0.00	6.00	0.00	5.00	1.00	15.00	2.00	18.00	9.00	19.00	1.00
Lift Pot	1	8.00	0.00	31.00	0.00	31.00	4.00	50.00	6.00	32.00	7.00	0.00	0.00
Move Can Pot	1	52.00	4.00	54.00	2.00	45.00	0.00	80.00	0.00	77.00	6.00	59.00	0.00
Move Pillbottle Pad	1	23.00	3.00	36.00	15.00	46.00	12.00	54.00	19.00	83.00	24.00	66.00	20.00
Move Playingcard Away	1	63.00	20.00	74.00	44.00	87.00	70.00	78.00	58.00	92.00	76.00	87.00	14.00
Move Stapler Pad	1	1.00	0.00	9.00	0.00	15.00	8.00	14.00	4.00	14.00	6.00	8.00	4.00
Open Laptop	1	66.00	18.00	81.00	24.00	88.00	51.00	88.00	51.00	92.00	60.00	14.00	3.00
Open Microwave	1	48.00	11.00	38.00	17.00	48.00	17.00	40.00	31.00	56.00	28.00	18.00	14.00
Pick Diverse Bottles	2	46.00	9.09	59.00	17.17	63.00	33.33	62.00	23.23	64.00	31.31	83.00	14.00
Pick Dual Bottles	2	52.00	34.00	75.00	39.00	71.00	54.00	63.00	40.00	86.00	45.00	86.00	20.00
Place A2b Left	1	29.00	6.00	53.00	6.00	32.00	8.00	57.00	27.00	62.00	17.00	64.00	12.00
Place A2b Right	1	18.00	4.00	43.00	11.00	41.00	6.00	51.00	32.00	54.00	18.00	59.00	6.00
Place Bread Basket	1	12.00	10.00	65.00	37.00	52.00	38.00	60.00	53.00	77.00	55.00	60.00	38.00
Place Bread Skillet	2	27.00	8.00	46.00	24.00	60.00	34.00	55.00	32.00	73.00	31.00	59.00	19.00
Place Burger Fries	2	67.00	62.00	82.00	70.00	92.00	86.00	82.00	84.00	81.00	73.00	66.00	45.00
Place Can Basket	2	41.00	3.00	39.00	5.00	55.00	11.00	54.00	3.00	65.00	21.00	53.00	7.00
Place Cans Plasticbox	2	21.00	19.00	53.00	28.00	59.00	50.00	58.00	50.00	84.00	68.00	28.00	27.00
Place Container Plate	1	81.00	39.00	87.00	43.00	83.00	42.00	88.00	58.00	88.00	57.00	90.00	58.00
Place Dual Shoes	2	19.00	5.00	35.00	16.00	68.00	35.00	60.00	37.00	63.00	19.00	46.00	3.00
Place Empty Cup	1	34.00	30.00	85.00	49.00	91.00	63.00	82.00	71.00	93.00	71.00	96.00	53.00
Place Fan	1	5.00	1.00	12.00	0.00	40.00	5.00	53.00	19.00	61.00	21.00	45.00	8.00
Place Mouse Pad	1	12.00	5.00	29.00	10.00	37.00	17.00	43.00	19.00	41.00	12.00	40.00	11.00
Place Object Basket	2	52.00	1.00	53.00	11.00	74.00	19.00	68.00	19.00	62.00	31.00	67.00	25.00
Place Object Scale	1	19.00	3.00	36.00	4.00	65.00	19.00	53.00	20.00	58.00	16.00	73.00	20.00
Place Object Stand	1	40.00	4.00	68.00	26.00	87.00	30.00	79.00	29.00	94.00	46.00	80.00	45.00
Place Phone Stand	1	15.00	6.00	34.00	14.00	45.00	18.00	62.00	29.00	50.00	22.00	54.00	7.00
Place Shoe	1	14.00	8.00	40.00	16.00	77.00	35.00	61.00	43.00	68.00	44.00	77.00	41.00
Press Stapler	1	71.00	39.00	85.00	62.00	84.00	64.00	78.00	56.00	82.00	58.00	61.00	72.00
Put Bottles Dustbin	3	17.00	9.00	40.00	14.00	70.00	56.00	64.00	40.00	59.00	50.00	63.00	10.00
Put Object Cabinet	2	28.00	4.00	50.00	5.00	49.00	6.00	57.00	12.00	61.00	12.00	34.00	7.00
Rotate QRcode	1	49.00	1.00	77.00	4.00	86.00	6.00	85.00	16.00	86.00	11.00	80.00	10.00
Scan Object	2	8.00	0.00	19.00	7.00	21.00	5.00	38.00	19.00	42.00	33.00	40.00	7.00
Shake Bottle	1	99.00	70.00	100.00	86.00	100.00	96.00	100.00	98.00	100.00	95.00	100.00	96.00
Shake Bottle Horizontally	1	100.00	76.00	100.00	92.00	100.00	94.00	100.00	98.00	100.00	94.00	100.00	94.00
Stack Blocks Three	3	5.00	2.00	10.00	1.00	48.00	11.00	46.00	19.00	31.00	7.00	54.00	14.00
Stack Blocks Two	2	38.00	6.00	55.00	13.00	93.00	39.00	94.00	37.00	62.00	21.00	85.00	33.00
Stack Bowls Three	3	52.00	23.00	74.00	42.00	74.00	44.00	66.00	42.00	68.00	44.00	70.00	24.00
Stack Bowls Two	2	84.00	51.00	92.00	75.00	90.00	59.00	88.00	72.00	91.00	63.00	93.00	52.00
Stamp Seal	1	12.00	8.00	39.00	17.00	57.00	25.00	38.00	26.00	61.00	20.00	48.00	15.00
Turn Switch	1	33.00	13.00	23.00	26.00	42.00	10.00	46.00	15.00	36.00	17.00	25.00	20.00
Average	–	**37.86**	**16.80**	**53.92**	**26.74**	**63.28**	**34.23**	**64.16**	**37.16**	**67.30**	**37.77**	**57.00**	**25.68**