

So You Think You Can Scale Up Autonomous Robot Data Collection?

Supplementary Material

Overview

We provide a brief overview of each appendix below. For videos, please see our website:
<https://sites.google.com/view/autonomous-data-collection>

Appendix A – Task Details

We give descriptions of each of our tasks, as well as more information on data scales and evaluation procedures.

Appendix B – Analyzing Human Supervision: Additional Results

We provide further details on the results from Section 4, including tables for all bar plots in the main text. We also include additional ablations on training method (training from scratch vs. fine-tuning) and additional experiments on training with autonomous data collected from out-of-distribution states.

Appendix C – Training Hyperparameters

We provide the training hyperparameters for all policies trained in this work.

A Task Details

In this section, we give additional information on the tasks studied in this work. We give verbal descriptions in [Appendix A.1](#), definitions of data scales in [Appendix A.2](#), and details on the evaluation procedures in [Appendix A.3](#).

A.1 Task Descriptions

- *FoldSock*. Fold a sock (with random configuration) neatly in half.
- *HangOvenMitt*. Hang an oven mitt (with random position and orientation) on a hook (fixed position).
- *HangTape*. Hang a roll of masking tape (with random initial position) on a hook (fixed position).
- *NutInsertion*. Insert a plastic nut (with random initial position) on a peg (fixed position).
- *Square*. Insert a square nut on a square peg (from [\[1\]](#)).
- *SoupInBasket*. Place a small soup can into a basket (from [\[2\]](#)).
- *BookInCaddy*. Place a book into a narrow book caddy (from [\[2\]](#)).
- *StackBowls*. Stack two bowls together and place both on a plate (from [\[2\]](#)).
- *RedMugOnPlate*. Put a red mug on a specific plate (from [\[2\]](#)).

We include an illustration of initial state distributions, sample initial and successful states, and sample camera observations for the NutInsertion and HangTape tasks in [Fig. 1](#).

A.2 Data Scale Definitions

For concision, and to focus on trends, we abbreviate data scales (i.e., number of demonstrations) as low (\downarrow), medium (\diamond), and high (\uparrow) for each of human demonstrations (H) and autonomous rollouts (A). Due to the fact that tasks vary widely in difficulty, the absolute value of demonstrations for each data scale varies per task. We include these values in [Table 1](#).

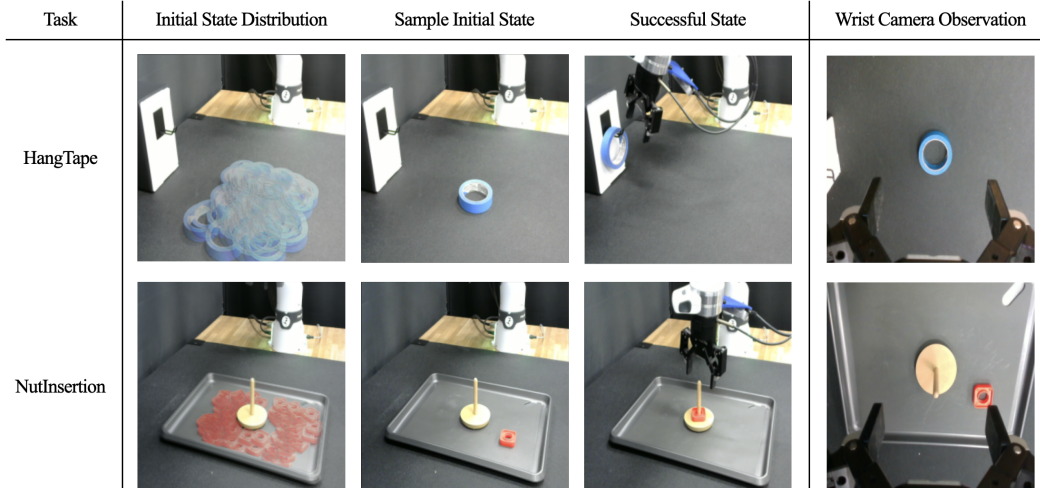


Figure 1: For the HangTape and NutInsertion tasks, we include scene images depicting the initial state distribution (using an overlay of initial state samples), a sample initial state, a successful state, and a view of the initial state from the wrist camera’s perspective.

Env	$\downarrow H$	$\diamond H$	$\downarrow A$	$\diamond A$	$\uparrow A$
HangTape	20	50	40	100	—
NutInsertion	50	100	100	—	—
Square	10	50	100	200	500
SoupInBasket	2	5	50	—	100
BookInCaddy	2	5	50	—	100
StackBowls	2	5	50	—	100
RedMugOnPlate	2	5	50	—	100

Table 1: Legend of data scales for each environment.

Example. To generate the training set for the $\downarrow H + \downarrow A$ setting on the NutInsertion task, we do the following:

- Collect 50 human demonstrations from randomly sampled initial states.
- Train an initial policy on the human demonstrations to convergence (approximately 47% success rate).
- Collect 100 successful autonomous rollouts (by rolling out the policy over 200 times and filtering out the failures).

A.3 Evaluation Procedure

Unless otherwise specified, all success rates in this work are calculated by uniformly sampling an initial state $s_0 \sim \rho_0$ and rolling out the learned policy under consideration until either a success state is achieved or a maximum time horizon is reached. For all simulation results, we perform 200 trials. For all real results, we perform 100 trials.

B Analyzing Human Supervision: Additional Results

In this section, we provide further details on the results in Section 4 of the main text. In [Appendix B.1](#), we ablate the choice of training from scratch on human-autonomous mixtures (the recipe used in all experiments in the main text). We also provide additional details on the results in Sections 4.1-4.2 regarding training with different data weights ([Appendix B.2](#)), data scales ([Appendix B.3](#)), number of rounds ([Appendix B.4](#)), and novelty-based reweighting ([Appendix B.5](#)). Finally, while experiments in the main text focus on autonomous data collected in-distribution, we provide additional experiments in [Appendix B.6](#) on training with autonomous data collected from out-of-distribution (OOD) scenarios.

53 B.1 Training from Scratch vs. Fine-tuning

54 All of the models trained on human-autonomous data mixtures in Section 4.1 are trained from scratch
55 until convergence. In this subsection, we justify this choice by comparing training from scratch to methods
56 involving fine-tuning.

57 Specifically, we focus on a single round of autonomous collection for the Square task in simulation.
58 Unless otherwise specified, each model is trained on a mixture of 50% autonomous, 50% human data.
59 We compare the following training recipes:

- 60 • *Scratch*: Train a new model from scratch on the human-autonomous mixture.
- 61 • *Fine-tune*: Fine-tune the autonomous policy checkpoint that generated the autonomous data on the
62 human-autonomous mixture.
- 63 • *Pre-train Autonomous + Fine-tune*: Pre-train a policy from scratch on the autonomous data only, and
64 then fine-tune on the human-autonomous mixture.
- 65 • *Scratch Add*: Directly aggregate human and auto data in one dataset (no explicit 50-50 sampling), and
66 train from scratch on this dataset.

67 In Table 2, we find that training from scratch, fine-tuning from the base policy, and training on combined
68 human and auto datasets all perform comparably. In fact, training methods seem to matter much less than
69 the amount of autonomous data provided. Therefore, for simplicity, we use the *Scratch* training method
70 for all other experiments in the main text.

Method	$\diamond H + \downarrow A$	$\diamond H + \diamond A$	$\diamond H + \uparrow A$
Scratch	69%	61.5%	79.5%
Fine-tune	68.5%	66%	67.5%
Pre-train Auto + Fine-tune	68.5%	69.5%	73.5%
Scratch Add	68.5%	66%	77.5%

Table 2: Comparing different training methods on Square in simulation, for medium amounts of human data ($\diamond H$) but for increasing amounts of autonomous data ($\downarrow A$ to $\diamond A$ to $\uparrow A$). All methods perform equivalently in each data regime.

71 B.2 Human and Autonomous Data Weights

72 Our experiments on Data Weights (Figure 5 in the main text) study the impact of relative sampling weights
73 of human-to-autonomous data. These experiments keep the amount of autonomous data fixed ($\downarrow A$) and
74 investigate if success rate changes for two scales of human data ($\downarrow H$ and $\diamond H$) at different sampling ratios
75 (75-25, 50-50, 25-75). We include these results in table form in Table 3 and Table 4. We find that changing
76 the sampling weights has almost no impact for a given data scale.

Env	$\downarrow H$ 75-25	$\downarrow H$ 50-50	$\downarrow H$ 25-75	$\diamond H$ 75-25	$\diamond H$ 50-50	$\diamond H$ 25-75
Square	15.5%	22%	21%	37.5%	38.5%	41%
SoupInBasket	39%	45.5%	41.5%	76%	83%	81.5%
BookInCaddy	34%	33%	34%	63.5%	61.5%	67%
StackBowls	57%	54%	52%	76%	81.5%	78.5%
RedMugOnPlate	75.5%	82.5%	80.5%	84%	86%	86%

Table 3: Different training weightings of human to autonomous data in simulation have negligible effects.

Env	$\downarrow H$ 75-25	$\downarrow H$ 50-50	$\downarrow H$ 25-75
HangTape	47%	55%	53%
NutInsertion	60%	57%	48%

Table 4: Different training weightings of human to autonomous data in real have negligible effects.

B.3 Human and Autonomous Data Scales

Our experiments on Data Scales (Figure 6 in the main text) use a 50-50 mixture and examine how success rate is impacted by the scale of initial human data and the ratio of human to autonomous data. We include the results in table form in Table 5. Including some amount of autonomous data tends to have mild positive effects in most cases, though these effects generally saturate as autonomous data scales. Increasing the scale of human data generally has a stronger effect than adding autonomous data.

Env	$\downarrow H$	$\downarrow H + \downarrow A$	$\downarrow H + \uparrow A$	$\diamond H$	$\diamond H + \downarrow A$	$\diamond H + \uparrow A$
Square	15.5%	22%	16%	44.5%	38.5%	43.5%
SoupInBasket	16.5%	33.5%	45.5%	54.5%	74%	83%
BookInCaddy	40.5%	30.5%	33%	51.5%	60%	61.5%
StackBowls	50.5%	59.5%	54%	83%	76%	81.5%
RedMugOnPlate	58%	80%	82.5%	79%	81.5%	86%
HangTape	44%	55%	48%	80%	80%	86%
NutInsertion	44%	57%	64%	53%	61%	—

Table 5: Scales of human data compared to autonomous data for 50-50 co-training on various simulation (top) and real (bottom) environments. More autonomous data often helps, but having more human data generally has a stronger effect.

B.4 Multiple Collection Rounds

Our experiments on Multiple Collection Rounds (Figure 7 in the main text) measure if any positive effects of autonomous data continue over multiple iterations. Specifically, we replace the autonomous data in the training mixture with the latest round of autonomous data collection, and re-train the model from scratch. The amount of autonomous data is kept constant at each round ($\diamond A$). We investigate the effects of multiple collection rounds at multiple scales of human data ($\downarrow H$ and $\diamond H$) in simulation and at the $\downarrow H$ scale in real. We present the results in table form in Table 6 and Table 8, generally observing plateaus in performance after an initial improvement in the first iteration. Interestingly, in the Square task, we observe a slight *decrease* in performance. Unlike the LIBERO tasks, Square contains a more challenging bottleneck state, and we hypothesize that subtle variations in the action distributions over multiple rounds of autonomous data collection and training may amplify this challenge. As evidence, in Table 7, we examine the “staged” success rate in Square over multiple iterations: note that the subtask for “moving the square” increases in success rate while the full task (which includes the insertion bottleneck) decreases in success rate.

Env	Base	Round 1 ($\diamond A$)	Round 2 ($\diamond A$)	Round 3 ($\diamond A$)	Round 4 ($\diamond A$)
Square ($\downarrow H$)	15.5%	17%	13%	21%	18.5
Square ($\diamond H$)	44.5%	38.5%	36%	35%	35%
SoupInBasket ($\downarrow H$)	12.5%	45.5%	60%	78%	—
SoupInBasket ($\diamond H$)	47%	84%	82.5%	82%	—
BookInCaddy ($\downarrow H$)	40%	40%	37.5%	44%	—
BookInCaddy ($\diamond H$)	45.5%	64%	74.3%	72%	—

Table 6: Multiple Rounds of autonomous collection using medium autonomous data ($\diamond A$) and training in simulation ($\downarrow H$ and $\diamond H$). We see either saturating increases or decreases in performance.

Stage	Base	Round 1 ($\diamond A$)	Round 2 ($\diamond A$)	Round 3 ($\diamond A$)	Round 4 ($\diamond A$)
Moves Square	67.5%	99.5%	100%	100%	94.5%
Full Success	44.5%	38.5%	36%	35%	35%

Table 7: Multiple Rounds of autonomous collection in Square ($\downarrow H$), illustrating the success rate for an intermediate stage (moving the square) and the full task.

Env	Base	Round 1 ($\diamond A$)	Round 2 ($\diamond A$)
HangTape ($\downarrow H$)	44%	55%	50%
NutInsertion ($\downarrow H$)	47%	57%	46%

Table 8: Multiple Rounds of autonomous collection using medium autonomous data ($\diamond A$) in real for HangTape and NutInsertion. We see that even though success rates improve in Round 1, they do not improve in Round 2.

96 B.5 Novelty-Based Reweighting Strategies

97 In Section 4.2, we consider if state novelty can be used as a proxy to extract more useful autonomous
 98 data, and form the basis for a sampling weight. In this section, we provide more details on these novelty
 99 measures. Given an ensemble of policies $\mathcal{E} = \{\pi_1, \pi_2, \dots, \pi_N\}$, we instantiate two measures of novelty
 100 building on ideas from prior work [3–5].

- 101 1. *Action Novelty*: Measure state novelty as proportional to the variance in the mean action
 102 predictions. This variance can be measured by an ensemble of policies trained on the same data:

$$\text{ActionNovelty}(s) = \sum_{i=1}^{N_A} \text{Var}_j(\mu_{ji})$$

103 where μ_j is the mean of the predicted action distribution $\pi_j(s)$ and N_A is the number of action
 104 dimensions.

- 105 2. *Embedding Novelty*: Measure state novelty as proportional to the variance in image embeddings
 106 produced by an ensemble of vision encoders (i.e., the encoders from each policy in \mathcal{E}):

$$\text{EmbeddingNovelty}(s) = \sum_{i=1}^{N_h} \text{Var}_j(h_{ji})$$

107 where $h_j = \text{enc}_j(s)$ (i.e., the embedding from the encoder associated with policy π_j) and N_h
 108 is the number of embedding dimensions.

109 Given a novelty measure, we assign the training weight for state s to be proportional to $\exp(\text{Novelty}(s)/\beta)$
 110 where β is a temperature hyperparameter.

111 B.6 Training on Out-of-Distribution Autonomous Successes

112 The experiments in the main text focus on training with autonomous data that is collected from
 113 *in-distribution* initial states (i.e., initial states are sampled from ρ_0 uniformly, or in the case of the active
 114 learning experiments, a reweighted version of ρ_0). In this section, we examine possible benefits from
 115 training on successful autonomous data from out-of-distribution (OOD) scenarios. More specifically, we
 116 generate the autonomous data by rolling out the initial policy from a new initial distribution ρ'_0 and collect
 117 autonomous successes which are the result of the policy generalizing to the new distribution.

118 In Table 9, we examine the impact on success rates when adding OOD autonomous data in the HangTape
 119 task. Specifically, we collect OOD autonomous data where one of two factors is varied compared to the
 120 initial distribution: the object (i.e., the tape is changed to a different roll of tape with a different color)
 121 and the distribution of initial object positions (i.e., the initial locations are sampled at an expanded outer
 122 boundary of the original distribution). When adding 50 successful autonomous rollouts from either of these
 123 OOD conditions to 50 in-distribution human demonstrations, we find positive impacts both in-distribution
 124 and in the OOD conditions. We see a similar trend in Table 10 on the NutInsertion task, where we collect
 125 autonomous data in OOD initial positions (i.e., the initial locations are from an expanded outer boundary)
 126 and find that both in-distribution and OOD performance improves.

127 These insights suggest that OOD autonomous data—i.e., successes that are the result of generalization
 128 in the initial policy—may be valuable, at the cost of potentially increasing environment design effort to
 129 change the initial state distribution of the environment.

Data Mixture	Success (ID)	Success (OOD Position)	Success (OOD Object)
50 H (ID)	80%	13%	27%
50 H (ID) + 50 A (OOD Position)	90%	23%	—
50 H (ID) + 50 A (OOD Object)	83%	—	51%

Table 9: Success rates both in-distribution (ID) and out-of-distribution (OOD) for policies trained on mixtures of in-distribution human data and OOD autonomous data on the HangTape task.

Data Mixture	Success (ID)	Success (OOD Object)
50 H (ID)	44%	40%
50 H (ID) + 50 (OOD Object)	52%	50%

Table 10: Success rates both in-distribution (ID) and out-of-distribution (OOD) for policies trained on mixtures of in-distribution human data and OOD autonomous data on the NutInsertion task.

130 C Training Hyperparameters

131 For all simulation experiments, we train using Diffusion Policy [6] with the following hyperparameters:

Diffusion Architecture	Conv1D UNet		
Prediction Horizon	16		
Observation History	2		
Num Action	8		
Kernel Size	5		
Num Groups	8		
Step Embedding Dim	256		
UNet Down Dims	[256, 512, 1024]		
Num Diffusion Steps	100		
Num Inference Steps	10		
Inference Scheduler	DDIM		
Observation Input	FiLM		
Image Encoder	ResNet-18		
Image Embedding Dim	256		
Proprioception	yes		

Training Steps	500000
Batch Size	64
Optimizer	AdamW
Learning Rate	1e-4
Weight Decay	1e-6
Learning Rate Schedule	Cosine Decay
Linear Warmup Steps	1000

Table 12: Training Hyperparameters, shared for all simulation experiments.

Table 11: Hyperparameters for Diffusion Policy, shared for all simulation experiments.

132 Our real-world experiments use the same hyperparameters, except with an observation history of 1, a
133 step embedding dimension of 128, and 2000 warmup steps. We train policies for the HangTape task for
134 400000 steps and policies for the NutInsertion task for 500000 steps.

135 References

- 136 [1] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu,
137 and R. Martín-Martín. What Matters in Learning from Offline Human Demonstrations for Robot
138 Manipulation. *arXiv:2108.03298*, 2021.
- 139 [2] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone. LIBERO: Benchmarking Knowledge
140 Transfer for Lifelong Robot Learning. In *Advances in Neural Information Processing Systems*
141 (*NeurIPS*), 2023.
- 142 [3] K. Gandhi, S. Karamcheti, M. Liao, and D. Sadigh. Eliciting Compatible Demonstrations for
143 Multi-Human Imitation Learning. In *Conference on Robot Learning*, 2022.

- 144 [4] R. Hoque, A. Balakrishna, E. R. Novoseller, A. Wilcox, D. S. Brown, and K. Goldberg. ThriftyDagger:
145 Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning. In *Conference on Robot*
146 *Learning*, 2021.
- 147 [5] S. Belkhale, Y. Cui, and D. Sadigh. Data Quality in Imitation Learning. In *Advances in Neural*
148 *Information Processing Systems (NeurIPS)*, 2023.
- 149 [6] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion Policy: Visuomotor
150 Policy Learning via Action Diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.