

Re-Mix: Optimizing Data Mixtures for Large Scale Imitation Learning

Anonymous Author(s)

Affiliation

Address

email

1 **Abstract:** Increasingly large robotics datasets are being collected to train larger
2 foundation models in robotics. However, despite the fact that data selection has been
3 of utmost importance to scaling in vision and natural language processing (NLP), little
4 work in robotics has questioned what data such models should actually be trained on.
5 In this work we investigate how to weigh different subsets or “domains” of robotics
6 datasets during pre-training to maximize worst-case performance across all possible
7 downstream domains using distributionally robust optimization (DRO). Unlike in
8 NLP, we find that these methods are hard to apply out of the box due to varying action
9 spaces and dynamics across robots. Our method, Re-Mix, employs early stopping and
10 action normalization and discretization to counteract these issues. Through extensive
11 experimentation on both the Bridge and OpenX datasets, we demonstrate that data
12 curation can have an outsized impact on downstream performance. Specifically, domain
13 weights learned by Re-Mix outperform uniform weights by over 40% on average and
14 human-selected weights by over 20% on datasets used to train the RT-X models.

15 **Keywords:** Data Curation, Data Quality, Robot Imitation Learning

16 1 Introduction

17 Many breakthroughs in machine learning can be attributed to “Internet-scale” datasets, from the
18 development of vision models like CLIP [1] to recent advancements in transformer-based language
19 modeling powered by the Common Crawl dataset [2]. Seeking to capitalize on this trend, several recent
20 efforts in robotics focus on collecting [3–6] or pooling [7] large scale robotics datasets with the goal of
21 training more performant imitation learning policies. Learning from this data, however, is particularly
22 challenging: robotics datasets are collected with different robots, environments, state spaces, action spaces,
23 and dynamics [8]. Moreover, actions themselves can be sub-optimal. Although the scale of robot data
24 is critical to replicating the success of large vision and language models, a fundamental question remains
25 in robotics: what data should actually be used for training?

26 While initial vision and language models were trained on highly-curated academic datasets like ImageNet
27 [9], questions surrounding data selection have shaped modern training pipelines that use internet scale data
28 [10–12]. For example, the training of large language models involves numerous stages of data filtering [13],
29 while large vision datasets like LAION [14] assess the quality of each data point using pre-trained models
30 like CLIP [1]. Thus as scaling of robot datasets continues, we can expect robotics data curation to become
31 equally critical. Unfortunately, simple filtering techniques are often inadequate in robotics; we cannot
32 apply n-gram filters, and visual embeddings do not capture the sequential nature of episodic robot data.

33 Even though aspects of demonstration data such as action quality [15] and visual diversity [3, 4, 16] have
34 been shown to be of paramount importance to downstream performance, approaches for robotics data
35 curation remain limited. In imitation learning, the data selection problem has only been characterized
36 theoretically [17, 18] or in small-scale toy settings [19]. Thus in practice we are left with ad hoc solutions:
37 as an example, the recently released Octo [20] generalist policy was trained on a curated subset of the
38 Open X-Embodiment dataset [OpenX; 7], a large collection of more than 60 individual robot datasets with
39 a total of more than 2M robot trajectories. The authors chose which datasets to include at what sampling

40 weight based on a subjective notion of “interestingness” [20]. The resulting data mix is aptly dubbed
41 the “magic soup,” owing to its mystical origins. RT-X [7] similarly determines dataset mixture weights
42 based on expert intuition. While the resulting data mixes are shown to work well in practice, their curation
43 requires extensive domain knowledge and manual data inspection. Such ad hoc selection strategies are
44 unlikely to scale to the rapidly growing datasets used to train robot policies [3, 5, 21].

45 In this work, we ask: how can we *automatically* curate large-scale robotics datasets to maximize perfor-
46 mance of generalist imitation learning policies across domains? Though many filtering techniques are not
47 directly applicable to robotics, we can borrow ideas from language modeling that systematically optimize
48 training data mixtures based on the model’s performance [22]. Specifically, we use group distributionally
49 robust optimization [23] to maximize the performance of a policy across all “domains” in a given dataset.
50 In the context of robotics, such “domains” can correspond to different scenes within a single dataset,
51 e.g., different toy kitchens for the Bridge V2 dataset [4], or full robot datasets in the case of multi-dataset
52 mixtures like the OpenX dataset. Unfortunately, naïvely applying such techniques [22] to robotics domains
53 results in poor performance due to the heterogeneity of robotics data in comparison to language. To fix these
54 problems, we propose Re-weighting Robotic Dataset Mixtures with Minimax Optimization (Re-Mix for
55 short). Re-Mix instantiates the data curation problem as a min-max optimization, where a policy *minimizes*
56 its excess behavior cloning loss over a reference model subject to learned domain mixture weights that try
57 to *maximize* it. Intuitively, the excess loss measures how much room the policy has to improve on a given
58 group, and our approach optimizes the data mixture to maximize such improvement potential. Crucially,
59 we carefully control the loss magnitudes between domains via action normalization and discretization,
60 even if the final policies we train are continuous diffusion models [24, 25]. Moreover, we find that careful
61 selection of the reference model is important to prevent overfitting, which can easily skew domain weights.

62 We empirically evaluate Re-Mix by using it to automatically optimize the training data mixture for the
63 Bridge V2 dataset [4] and the OpenX-based dataset used to train RT-X [7]. We show that policies trained
64 with our data mix improve performance by 40% and 20% respectively over naïve data balancing and
65 human-expert-curated data mixtures in evaluations across WidowX and Franka robot arms. Additionally,
66 we show that weights from Re-Mix can effectively *sub-sample* both datasets, achieving competitive
67 performance using only 25% of the original data, while using uniform or human curated weights
68 significantly reduces performance.

69 Our contributions are as follows:

- 70 • We introduce Re-Mix, which extends Xie et al. [22] for automatically curating large-scale robotics
71 datasets using group distributionally robust optimization.
- 72 • We demonstrate Re-Mix’s ability to curate effective training mixes for the Bridge [4] and RT-X [7] data.
- 73 • We release curated subsets of the Bridge and RT-X datasets, which can be used for training generalist
74 policies with minimal loss in performance, while significantly reducing the required compute budget.

75 2 Related Work

76 In congruence with the rise of deep learning in various fields, data selection has become of increasing
77 interest. Here we review the most relevant works, organized by area.

78 **The Data Problem in Robotics.** Several recent works in robotics have focused on collecting large demon-
79 stration datasets for imitation learning in simulation [15, 26, 27] and the real world [3, 7, 28–32] to train
80 large-scale robot policies [6, 20, 33, 34]. Generally, these works along with others that study the influence
81 of data collection on compositional generalization [16, 35, 36] show that aspects of dataset construction
82 such as scene and task diversity have a direct impact on downstream policy generalization. Though several
83 studies focus on *how* data should be collected via specific hardware [37], collection procedures [16, 38, 39],
84 or provide theoretic insights about data collection [17], little work in robotics addresses the post-hoc dataset
85 selection and analysis problem. This is particularly important as the number and diversity of robot datasets
86 are increasing with less clear conclusions about how to train a policy that effectively consumes all the col-
87 lected data [3, 7, 20]. Baker et al. [40] train a classifier to predict data quality, but require human annotations
88 which are impractical to scale. Perhaps most related are retrieval-based methods that subset datasets [41, 42],
89 but do so based on a priori target task specifications and are thus inapplicable to training generalist policies.

90 **Data Curation in Computer Vision.** Computer vision datasets were originally hand-crafted and manually
 91 labeled [9, 43]. However, tapping into more high quality data has been crucial to increasing performance
 92 [1, 44]. Notably, filtering techniques based on metadata-count balancing [45], embeddings [14], optical
 93 flow [46], and clustering [47] have shown to greatly improve downstream performance despite filtering
 94 out large amounts of data. Taking it to the extreme, coreset selection methods select miniscule subsets
 95 of vision datasets using active learning [48, 49], but on small datasets due to computational requirements
 96 [50, 51]. Though learning from demonstrations may involve vision, at its core is *action* prediction. Data
 97 curation techniques from computer vision can only filter state-action trajectories in an action-agnostic
 98 manner – potentially removing useful parts of a dataset.

99 **Data Curation in Natural Language Processing.** When training on large real-world sources of text,
 100 language modeling pipelines employ a number of text-specific preprocessing steps including metadata
 101 filtering, language filtering, de-duplication, and toxicity reduction [13]. More advanced methods for data
 102 selection consider sub-setting data to maximize downstream performance, as we also do in this work, but
 103 use techniques such as k-means clustering over embedded text [52, 53]. While such clustering techniques
 104 can potentially be visually informative in robotics as well – similar to curation works in computer vision
 105 – they do not provide information about *actions*. Mixture techniques, such as Domain Reweighting with
 106 Minimax Optimization (DoReMi) [22] balance text domains using robust optimization and build upon
 107 ideas from prioritized training [54–56]. Our work is inspired by DoReMi as such robust optimization
 108 approaches can be applied to imitation learning as well. However, directly applying DoReMi to robotics
 109 datasets is non-trivial as we show in the next section.

110 3 Re-weighting Robotic Dataset Mixtures with Minimax Optimization

111 In this section, we first formalize the problem of re-weighting robotics data mixtures for imitation learning.
 112 We then discuss our approach which uses distributionally robust optimization for selecting domain weights
 113 and sub-setting large robotics datasets.

114 **Problem Setup.** We consider the general imitation learning problem, where we are given a dataset of
 115 demonstrations $\mathcal{D} = \{\tau_1, \dots, \tau_n\}$ consisting of state-action trajectories $\tau = (s_1, a_1, \dots, s_{T_i}, a_{T_i})$. Our goal is
 116 to learn a parameterized policy π_θ that learns a mapping from states to actions $\pi_\theta : \mathcal{S} \rightarrow \mathcal{A}$. In practice, this
 117 is often done through standard imitation learning algorithms such as behavior cloning (BC) by minimizing
 118 the expected negative log-likelihood of the actions under the policy:

$$\mathcal{L}_{\text{BC}}(\pi_\theta, \mathcal{D}) = \mathbb{E}_{(s,a) \sim \mathcal{D}}[-\log \pi_\theta(a|s)] \quad (1)$$

119 However, datasets often contain more information than just state action pairs. We assume that the overall
 120 dataset \mathcal{D} can be split into k heterogeneous domains $\mathcal{D}_1, \dots, \mathcal{D}_k$. This is a general assumption: while these
 121 domains could be larger groups, like different datasets from the Open X-Embodiment dataset [7], they
 122 could also be as small as single trajectories. Moreover, each of the k domains can differ in state space
 123 \mathcal{S} , action space \mathcal{A} , transition dynamics, or their distributions. In fact when learning large behavior models,
 124 such heterogeneity becomes necessary to access more sources of diverse data. In this work, we use the
 125 Bridge dataset [4] – with different environments as the domains, and the Open-X-Embodiment dataset [7]
 126 – with different robot embodiments as the differing domains.

127 Our goal is to learn a weighting vector $\alpha \in \Delta^k$ that specifies a probability distribution over all domains
 128 such that any model, when trained on a domain mixture weighted according to α , attains maximum
 129 performance *across all domains*. We note that unlike the data retrieval problem, which aims to curate data
 130 *for a particular target task*, our goal is to curate datasets for effective pre-training or co-training without
 131 any a priori knowledge of a target task.

132 **Distributionally Robust Optimization.** When pre-training on large amounts of robot data we want
 133 policies to *generalize* to new settings and tasks, not master a specific target task. With that in mind,
 134 we want to optimize for a data mixture that results in models that i) can perform as well as possible on
 135 each domain, but ii) do not overfit to any one domain at the expense of another. Distributionally robust
 136 optimization (DRO) techniques aim to solve the same problem: learn models that minimize the worst-case
 137 training loss [23] – BC loss in the case of imitation learning – across domains $\mathcal{D}_1 \dots \mathcal{D}_k$. Specifically,

138 naively applying group robust optimization techniques in robotics would result in the following objective:

$$\min_{\theta} \max_{\alpha \in \Delta^k} \sum_{i=1}^k \alpha_i \mathcal{L}_{\text{BC}}(\pi_{\theta}, \mathcal{D}_i). \quad (2)$$

139 With this objective, α up-weights domains that have a higher loss value, emphasizing the hardest domains.
 140 However, in practice we might not be interested in just fitting the domains with higher losses. For example,
 141 a robotics dataset with complex multi-modal rotation movements for bottle-cap unscrewing might always
 142 have higher BC loss than simple pick-place datasets. Thus, standard robust optimization techniques could
 143 end up ignoring the latter domain. Instead, as in prior work [22, 48, 57] we consider the *difference* in loss
 144 between our learned policy π_{θ} and a reference policy π_{ref} which is trained to convergence on a data mix that
 145 uniformly samples all domains. In Eq. (2) this equates to replacing \mathcal{L}_{BC} with $\mathcal{L}_{\text{BC}}(\pi_{\theta}, \mathcal{D}_i) - \mathcal{L}_{\text{BC}}(\pi_{\text{ref}}, \mathcal{D}_i)$.
 146 We refer to this difference as the *excess loss*, and use it for robust optimization. Like before, this will
 147 down-weight domains that the policy fits well, as it can achieve a loss similar to that of the reference model.
 148 However, it crucially also down-weights domains which are difficult to fit (i.e. they have a high policy and
 149 reference loss) due to the relative nature of the excess loss. This can happen in the presence of sub-optimal
 150 actions. Only domains that have a high excess loss, meaning the policy can improve to match the reference,
 151 will be up-weighted as α is chosen to maximize the excess overall loss.

152 Unfortunately, models learned directly using robust optimization often exhibit worse overall performance
 153 [58, 59]. Instead, we can use the learned α vector for downstream training as in Xie et al. [22]. This gives us
 154 a set of reusable weights that can be used to train different policies without the need for robust optimization.

155 3.1 The Challenges of Applying Robust Optimization in Robotics

156 While Group DRO has been applied in language modeling, robust optimization techniques face unique chal-
 157 lenges in robotics which we highlight here. In the next section, we detail how we incorporate the following
 158 changes into a distributionally robust optimization pipeline to select domain weights for robotics datasets.

159 **Unbalanced Losses.** Large robotics datasets are often highly hetero-
 160 geneous: many are collected across different embodiments, controllers,
 161 and frequencies or even different units (e.g., inches vs meters). Even
 162 within the same dataset, different scenes or tasks require different types of
 163 actions. Consequently, action magnitudes can be vastly different, causing

	α_{noise}	α_{bridge}
Bounds	0.943	0.057
Gaussian	0.158	0.842

Table 1: Learned α from toy setting in Section 3.1

164 some datasets to have an outsized effect on robust optimization. To demonstrate this, we construct a toy
 165 experiment where we compare training a policy with Group DRO [23] when the action distributions match
 166 to when they differ. Specifically, we assign Gaussian random actions to half of the trajectories in the
 167 Bridge V2 dataset [4] (*random*). On the other half (*bridge*), we use the original actions, either standardized
 168 to a Gaussian to match the noise distribution, or re-scaled between -1 and 1 (“Bounds” normalization [24]).
 169 We show the learned α for each scheme in Table 1. While one might expect that α would correctly assign
 170 majority weight to the “bridge” domain since the “noise” domain is impossible for both the policy and
 171 reference model to fit, this is actually only true in the “Gaussian” case when the action distributions of both
 172 domains are more aligned. When using “Bounds” normalization, the average action magnitude is lower, and
 173 thus the losses on the *bridge* domain are dwarfed by the *random* domain. To mitigate this issue, we apply
 174 Gaussian normalization to each domain independently when learning from heterogeneous action spaces.

175 **Continuous Losses.** Robust optimization has largely been applied in discrete classification problems
 176 with cross-entropy losses, for example in language modeling [57]. Popular policy learning approaches,
 177 however, often predict continuous actions and use L1 or L2 objectives [15, 24, 60, 61]. This is problematic
 178 for two reasons. First, action distributions can be multi-modal, and expressive continuous policy classes
 179 like diffusion [24] only optimize an upper bound on the true loss. Second, robot datasets often have a
 180 large number of action outliers, which can heavily sway the value of continuous action losses. Using
 181 a continuous loss function would heavily influence loss magnitudes for domains with multi-modal or
 182 outlier actions which for the aforementioned reasons is problematic for DRO. To resolve this problem,
 183 we discretize each action dimension via binning when using robust optimization.

184 **Overfitting.** Datasets in language modeling often contain billions of tokens. As a result, methods like
 185 Xie et al. [22] do not observe overfitting because of the scale of their training data. On the other hand,

186 large robot datasets are comparatively small (~ 10 -100k demonstrations). Moreover, individual datasets
 187 in mixtures like the Open X-Embodiment dataset [7] can be as small as 100 demonstrations. In this regime,
 188 it is common for particularly high-capacity policies to achieve near-zero training loss for every datapoint
 189 [6, 20, 62]. This is problematic when using the excess loss for robust optimization: if the reference model
 190 achieves near-zero training loss on every data point within a domain, the excess loss is equivalent to the
 191 regular loss (since the reference loss is always $\simeq 0$) and α no longer reflects the potential for improvement
 192 on each domain. To counteract this problem, we employ aggressive early stopping on both the reference
 193 model and robust optimization. Specifically, we select the latest checkpoint from the reference model
 194 that has not overfit to *any* of the domains $\mathcal{D}_1, \dots, \mathcal{D}_k$ as measured by the difference in loss values between
 195 the training dataset and a held-out validation dataset for the respective domain. We then run robust
 196 optimization for the same number of steps. This ensures that the reference model does not overfit to any
 197 individual domain and the learned weights α are informative.

198 3.2 Re-weighting Robotic Dataset Mixtures with Minimax Optimization

199 Our final approach, Re-Mix, uses Group distributional robustness to determine the weights of policy
 200 training data mixture and incorporates the key design considerations from the previous section.

201 **Stage 1: Action Preprocessing.** Following Section 3.1 we apply Gaussian normalization separately to
 202 every domain with different action spaces and dynamics, and then discretize actions via binning.

203 **Stage 2: Reference Model Training.** Next, we train a discrete reference model π_{ref} on the uniform mixture
 204 of domains $\mathcal{D}_1, \dots, \mathcal{D}_k$, where each domain is weighted in proportion to its size. We select the final reference
 205 model checkpoint by validation loss per Section 3.1, and use it to estimate the excess loss per domain.

206 **Stage 3: Group Distributionally Robust Optimization.** We learn the domain weights α via the following
 207 robust optimization with a discrete policy π_θ :

$$\min_{\theta} \max_{\alpha \in \Delta^k} \sum_{i=1}^k \alpha_i \left[\frac{1}{|\mathcal{D}_i|} \sum_{(s,a) \in \mathcal{D}_i} (-\log \pi_\theta(a|s) + \log \pi_{\text{ref}}(a|s)) \right] \quad (3)$$

208 which minimizes the worst case excess BC loss of the learned policy $-\log \pi_\theta(a|s) + \log \pi_{\text{ref}}(a|s)$ over
 209 all possible weightings of the domains $\alpha \in \Delta^k$. To update α we use the GroupDRO optimizer [23], which
 210 preforms one step of exponentiated gradient ascent on α followed by domain-weighted gradient descent
 211 on θ at each training step. Our resulting values of α upweight domains that we can still improve on, while
 212 downweighting domains that are trivial or impossible to fit. This means that Re-Mix directly filters data
 213 based on actions, unlike other techniques in vision and language that solely filter based on embeddings.
 214 We optimize Eq. (3) for the same number of steps as the reference model.

215 **Stage 4: Data Weighting for Policy Training.** After our robust optimization stage over the excess loss,
 216 we take the average value of α over the course of training, which we denote by $\bar{\alpha}$. We can then use this
 217 value of $\bar{\alpha}$ to re-weight different domains, or even subset datasets for co-training.

218 4 Experiments

219 We aim to answer the following questions: (1) Does Re-Mix effectively curate large robot datasets for down-
 220 stream policy learning? (2) Can we use Re-Mix to heavily sub-sample robot datasets while retaining good
 221 performance? (3) Which design decisions matter for effective automatic curation of large robot datasets?

222 4.1 Experimental Setup

223 **Datasets.** We test Re-Mix curation on two widely-used, large-scale robot datasets: (1) the Bridge V2
 224 Dataset [4], consisting of 50k diverse teleoperated demonstrations of single-arm manipulation tasks with a
 225 WidowX 6DoF robot arm, and (2) the datasets from the Open X-Embodiment dataset used to train RT-1-X
 226 and RT-2-X models [7], consisting of a total of 350k demonstrations which span disparate embodiments
 227 and environments. We partition the Bridge V2 dataset into 32 domains $\mathcal{D}_1, \dots, \mathcal{D}_{32}$ based on the scenes the
 228 data was collected in. For OpenX, we use each of the 11 datasets in the RT-X training set as a domain for
 229 our curation experiments. The OpenX data mix is particularly challenging for effective curation due to its
 230 heterogeneous data sources. For a detailed list of all datasets and partitions, see Appendix B.

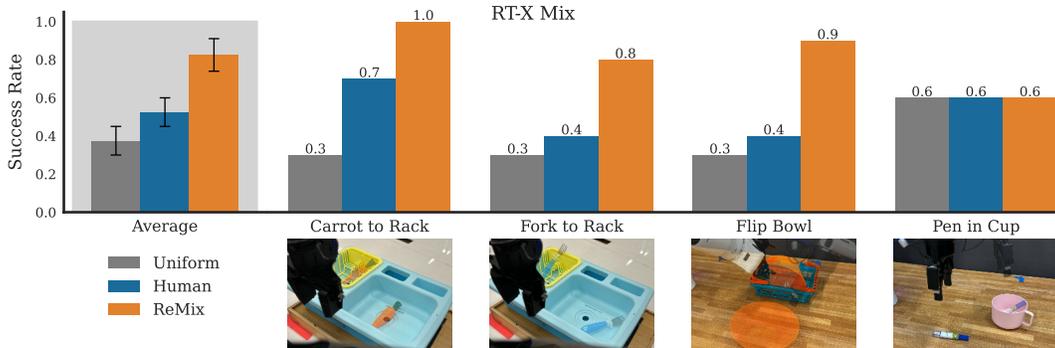


Figure 1: Results for curating the RT-X training mix. We test policies trained on different weightings of the RT-X data mixture across two WidowX and two Franka tabletop manipulation tasks. We find that the policy trained on the data mix curated with Re-Mix achieves strongest performance, even outperforming the human-expert-curated data mix from RT-X [7]. Mean \pm StdErr across 4 tasks, 10 evaluations each.

231 **Training and Evaluation Details.** We aim to assess the quality of various curated pre-training data
 232 mixtures for downstream policy learning. To that end, we co-train generalist goal-conditioned policies on
 233 the curated datasets. As we do not have access to the robot setups used to collect the datasets we train on, we
 234 construct our own WidowX and Franka robot evaluation setups. Unfortunately, policies trained on *only* the
 235 pre-training data failed to zero-shot generalize to our out-of-distribution setups. To address this, we follow
 236 prior works [3, 41, 42, 63] and co-train our policies on a small amount of in-domain data (25 demonstrations
 237 each for 3 representative tasks), added to the final training mixture at a small weight of 5%. We then evaluate
 238 policies on tasks that are out-of-distribution with respect to the co-training data to test generalization. As a
 239 result of co-training, all policies achieve non-zero success rate. Importantly, the in-domain dataset is small
 240 enough that the quality of the pre-training data mix still has significant impact on the evaluation result, pro-
 241 viding a good test bed for data curation approaches. All models are evaluated in the real world with 10 trials
 242 per task totaling over 400 real-world trials cumulatively. For all policies we use a ResNet 50 image encoder
 243 [64]. For the Re-Mix reference model and Group DRO optimization we use a discrete MLP action head. For
 244 all final policies we use the diffusion head from [4, 20, 65] and train all models for 400,000 gradient steps.

245 **Comparisons.** We compare the quality of Re-Mix’s curated data mixes to a naïve baseline: sampling
 246 uniformly from each domain according to the total number of state-action pairs (**Uniform**). For evaluations
 247 on the OpenX datasets, we additionally compare to a human-expert-curated data mix, using the hand-crafted
 248 weights from RT-X [7]. For Bridge there is no expert-curated data mix — uniform sampling is the norm.

249 4.2 How do Re-Mix weights impact performance?

250 In Fig. 1 we show results for weighing datasets from the RT-X mix
 251 according to different methods. We consider four tasks, two on a
 252 WidowX robot and two on a Franka Panda. On the WidowX, the “Carrot
 253 to Rack” task tests generalization to an unseen object while “Fork to
 254 Rack” tests generalization to unseen initial conditions. On the Panda, the
 255 “Pen in Cup” task also tests generalization to unseen initial conditions,
 256 while the “Flip Bowl” task tests performance on motions not seen in the
 257 pre-training data. Additionally, our Panda robot uses a Robotiq 2F-85
 258 gripper, which was not present in any of the RT-X-datasets. Note that
 259 for the RT-X mix, we co-train the same model on both the WidowX and Franka data. As expected, we find
 260 that the human-expert-curated dataset weights outperform the naïve uniform sampling baseline by 20%.
 261 Qualitatively, we find that all models perform similarly on the “Pen in Cup” task, but the policy trained
 262 on the Re-Mix data mixture performs particularly well on “Flip Bowl.” We hypothesize that this is because
 263 Re-Mix up-weights the relatively small “Toto” which contains similar pouring motions to the flip bowl task.

264 Fig. 2 shows results using Re-Mix weights versus uniform weighting over scenes in the Bridge dataset. We
 265 find that performance in this setting is similar across both models. We posit that in the presence of the full
 266 bridge dataset, selecting weightings is less important as the model is able to fit every scene well.

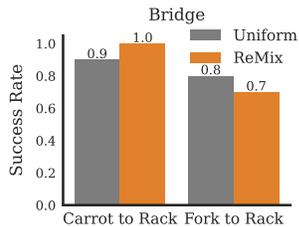


Figure 2: On Bridge V2 [4] there is no notable difference between uniform sampling vs. Re-Mix when training on the full dataset.

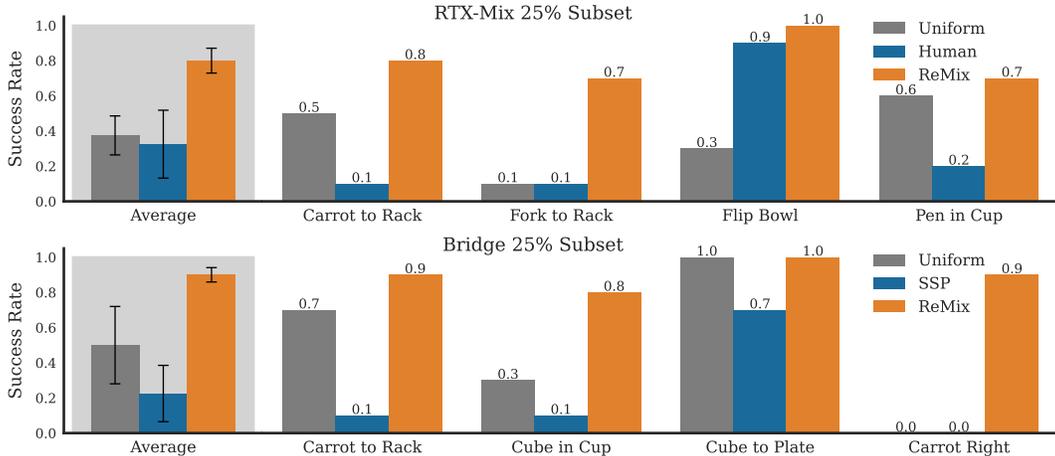


Figure 3: Results sub-setting datasets via different strategies until they reach 25% of their original size. We again use 10 evaluations per task, and show the Mean \pm StdErr.

267 4.3 How well does Re-Mix subset datasets?

268 Though co-training on diverse data is important for performance [3, 63], doing so is often expensive
 269 given that modern robot datasets like the Open X-Embodiment dataset encompass TBs of data. In this
 270 section, we evaluate how well Re-Mix can be used to *subset* datasets. The key idea: if Re-Mix weights
 271 are proportional to the importance of the data in each domain, we can use them to effectively sub-set
 272 the dataset by removing data from domains that Re-Mix assigns low weight. If doing so retains policy
 273 performance, we can substantially decrease the required storage and compute for training policies locally.

274 We subset the base datasets according to Re-Mix and baselines by first computing the target size of
 275 the entire data mix *after* sub-setting, in our case 25% of $|\mathcal{D}|$. Then, we remove datapoints according to
 276 the mixture weights $\bar{\alpha}$. If a small dataset is upweighted too much (i.e. if domain i is 1% of $|\mathcal{D}|$ but is
 277 upweighted to $\bar{\alpha}_i = 5\%$), there might not be enough data to exactly match $\bar{\alpha}$ from subsetting alone (even
 278 if we take all data from the 1% domain, the most it can make up of a 25% subset is 4%). Thus, we sample
 279 extra points from the datasets uniformly and at training still weight datasets according to the exact $\bar{\alpha}$.

280 Good subsetting methods ought to retain performance when data is removed. Again, we compare
 281 performance of Re-Mix to using naïve uniform sampling for subsetting, and to subsetting based on the
 282 human expert weights. For Bridge, where no expert weighting exists, we additionally compare to a vision
 283 and language subsetting method called “Self-Supervised Prototypes” (SSP) [66] which runs k-means on
 284 image embeddings and discards data closest to each centroid. We average CLIP embeddings across each
 285 trajectory to obtain the embeddings for k-means. As subsetting data induces a larger performance gap
 286 than simple domain-weighting on Bridge, we design additional tasks to assess performance. “Cube to
 287 Plate” and “Cube in Cup” introduce distractors in goal images, while “Carrot to Right” evaluates a motion
 288 unseen in co-training.

289 Our subsetting results can be found in Fig. 3. Overall, we find that subsetting exacerbates the difference
 290 between methods, as the weights now directly affect dataset composition. On RTX, Re-Mix retains
 291 performance with 25% versus 100% (80% vs 82.5% on average). This is likely because Re-Mix places
 292 higher weights on some of the smaller datasets and down-weights the large Kuka dataset from [34]. For
 293 example, Re-Mix retains 72% of the Berkeley UR5 Dataset and only 12% of the Kuka dataset, while the
 294 human weighting scheme retains only 30% of the UR5 Dataset and 24% of Kuka. On Bridge, Re-Mix also
 295 retains performance (90% vs. 100% on “Carrot to Rack”) and out-performs other methods on the new tasks
 296 by a large margin. Overall SSP performs poorly, likely since robot trajectories may be out-of-distribution
 297 for vision models like CLIP and thus return similar embeddings for all images containing the same robot,
 298 regardless of scene differences. Appendix A includes additional results for Bridge with 10% subsetting.

299 4.4 Analyzing Re-Mix Weights

300 Table 2 shows the weights produced by different methods on the RT-X dataset mix in comparison to
 301 the uniform weights corresponding to the original size of the respective dataset (as fraction of the total

Method	α_{UR5}	$\alpha_{\text{Cable Routing}}$	α_{Bridge}	α_{Jaco}	α_{Kuka}	α_{RoboTurk}	α_{RT1}	$\alpha_{\text{Taco Play}}$	$\alpha_{\text{Taco Extra}}$	α_{Toto}	α_{Viola}
Uniform	1.01%	0.43%	22.7%	0.81%	24.9%	1.94%	40.9%	0.60%	2.46%	3.42%	0.80%
Human	1.22%	1.56%	27.5%	1.95%	25.1%	2.35%	26.8%	1.46%	5.94%	4.13%	1.90%
Re-Mix	2.37%	0.20%	19.9%	0.39%	12.1%	1.14%	42.5%	0.63%	3.04%	16.3%	1.51%

Table 2: Dataset mixture weights by different methods on RT-X. We color relative increases of more than 25% from uniform green and color relative decreases of more than 25% red.

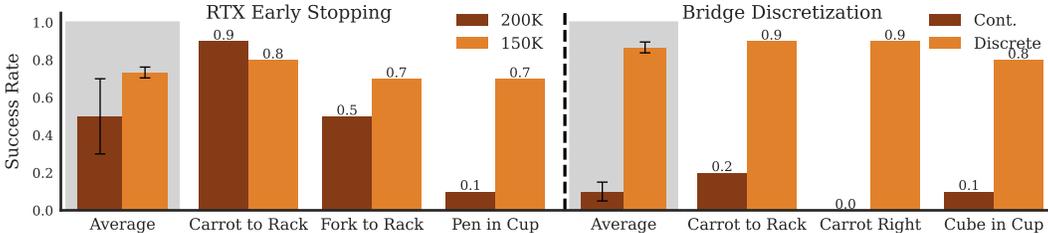


Figure 4: Ablations for design choices in Re-Mix. We ablate the effects of **left**: reference model overfitting by selecting a checkpoint once validation loss starts increasing at 150K steps and **right**: using continuous actions for Re-Mix. For ablations, we remove the ‘Flip Bowl’ and ‘Cube to Plate’ tasks as all Re-Mix variants achieved 100% success.

302 dataset). The human-expert-designed weights largely down-weight RT-1 [6], while up-weighting some
 303 of the smaller datasets like Routing [67], and Taco [68]. On the other hand, Re-Mix largely down-weights
 304 the Kuka dataset [69]. This dataset was autonomously collected and then filtered by success, making
 305 it of potentially lower action quality. Re-Mix also down-weights some smaller domains that are easy
 306 to fit; for example, Cable Routing has no gripper actions and Jaco [70] only has three possible actions.
 307 Surprisingly, Re-Mix up-weights the Toto dataset [71] by more than 4x. We posit that this is because Toto
 308 has a particularly multi-modal action distribution which deviates far from a standard Gaussian even after
 309 normalization and thus may be more challenging to fit. See Appendix A for a plot of its action distribution.

310 4.5 What matters in Re-Mix?

311 In this section we ablate several design choices used in Re-Mix (see Section 3.1), including action
 312 discretization and early stopping. We run all ablations in the 25% subset setting (see Section 4.3), since
 313 subsetting further amplifies the effects of the domain weights. In Fig. 4 we first analyze the effects of
 314 choosing a reference model checkpoint for Group DRO that is overfit to the training dataset. Empirically,
 315 we find that choosing a checkpoint just 50K steps after early stopping decreases performance by over
 316 15% on average, likely because the reference model baseline used to determine the domain weights is
 317 less meaningful once it overfits. On the right half of Fig. 4, we show performance on Bridge when using
 318 continuous (Cont.) actions in Re-Mix instead of discrete for estimating α . We find that continuous actions
 319 lead to significantly worse performance, as their loss functions fail to fit outliers or multi-modal actions.

320 5 Limitations and Future Work

321 In this work we present Re-Mix, a method for automatically curating robotics datasets using distributionally
 322 robust optimization.

323 **Evaluation.** While we train on large, diverse robot datasets, the need for real world trials makes it difficult
 324 to exhaustively evaluate trained generalist policies on many robot embodiments and setups. While our
 325 evaluations capture two widely used robot arms from prior works [4, 7, 20], WidowX and Franka, future
 326 work should extend to more embodiments, perhaps via simulated environments [62].

327 **Abnormal Action Distributions.** We have noticed that our method upweights datasets with abnormal
 328 action distributions such as the Toto dataset. While resulting data mixes performed well, such up-weighting
 329 is not necessarily desirable. We hope to achieve less sensitivity to such irregularities in future work.

330 **Computational Cost.** Using our pre-computed weights can significantly reduce the compute required to
 331 train generalist policies. However, our approach for computing Re-Mix weights requires training policies
 332 on the full data twice. Future work can instead strive to curate datasets ‘on-the-fly’ within one run.

333 **Scaling Up.** While we have demonstrated improvements on two large datasets, Bridge V2 and RT-X,
 334 scaling up to even larger ones such as the Open X dataset [7] (>2M episodes) is an exciting extension.

335 **References**

336 [1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin,
337 J. Clark, et al. Learning transferable visual models from natural language supervision. In *International*
338 *conference on machine learning*, pages 8748–8763. PMLR, 2021.

339 [2] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott,
340 L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv*
341 *preprint arXiv:1911.02116*, 2019.

342 [3] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K.
343 Srirama, L. Y. Chen, K. Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset.
344 *arXiv preprint arXiv:2403.12945*, 2024.

345 [4] H. Walke, K. Black, A. Lee, M. J. Kim, M. Du, C. Zheng, T. Zhao, P. Hansen-Estruch, Q. Vuong,
346 A. He, V. Myers, K. Fang, C. Finn, and S. Levine. Bridgedata v2: A dataset for robot learning at
347 scale. In *Conference on Robot Learning (CoRL)*, 2023.

348 [5] H.-S. Fang, H. Fang, Z. Tang, J. Liu, J. Wang, H. Zhu, and C. Lu. Rh20t: A robotic dataset for
349 learning diverse skills in one-shot. In *RSS 2023 Workshop on Learning for Task and Motion Planning*,
350 2023.

351 [6] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman,
352 A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. Joshi, R. Julian,
353 D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch,
354 O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar,
355 P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega,
356 Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-1: Robotics transformer for
357 real-world control at scale. In *arXiv preprint arXiv:2212.06817*, 2022.

358 [7] Open X-Embodiment Collaboration, A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan,
359 A. Khazatsky, A. Rai, A. Singh, A. Brohan, A. Raffin, A. Wahid, B. Burgess-Limerick, B. Kim,
360 B. Schölkopf, B. Ichter, C. Lu, C. Xu, C. Finn, C. Xu, C. Chi, C. Huang, C. Chan, C. Pan, C. Fu,
361 C. Devin, D. Driess, D. Pathak, D. Shah, D. Büchler, D. Kalashnikov, D. Sadigh, E. Johns, F. Ceola,
362 F. Xia, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Schiavi, H. Su, H.-S. Fang,
363 H. Shi, H. B. Amor, H. I. Christensen, H. Furuta, H. Walke, H. Fang, I. Mordatch, I. Radosavovic,
364 I. Leal, J. Liang, J. Kim, J. Schneider, J. Hsu, J. Bohg, J. Bingham, J. Wu, J. Luo, J. Gu,
365 J. Tan, J. Oh, J. Malik, J. Tompson, J. Yang, J. J. Lim, J. Silvério, J. Han, K. Rao, K. Pertsch,
366 K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka,
367 K. Zhang, K. Majd, K. Rana, K. Srinivasan, L. Y. Chen, L. Pinto, L. Tan, L. Ott, L. Lee, M. Tomizuka,
368 M. Du, M. Ahn, M. Zhang, M. Ding, M. K. Srirama, M. Sharma, M. J. Kim, N. Kanazawa, N. Hansen,
369 N. Heess, N. J. Joshi, N. Suenderhauf, N. D. Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer,
370 P. R. Sanketi, P. Wohlhart, P. Xu, P. Sermanet, P. Sundareshan, Q. Vuong, R. Rafailov, R. Tian,
371 R. Doshi, R. Martín-Martín, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani,
372 S. Levine, S. Moore, S. Bahl, S. Dass, S. Song, S. Xu, S. Haldar, S. Adebola, S. Guist, S. Nasiriany,
373 S. Schaal, S. Welker, S. Tian, S. Dasari, S. Belkhale, T. Osa, T. Harada, T. Matsushima, T. Xiao,
374 T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, V. Jain, V. Vanhoucke, W. Zhan,
375 W. Zhou, W. Burgard, X. Chen, X. Wang, X. Zhu, X. Li, Y. Lu, Y. Chebotar, Y. Zhou, Y. Zhu,
376 Y. Xu, Y. Wang, Y. Bisk, Y. Cho, Y. Lee, Y. Cui, Y. hua Wu, Y. Tang, Y. Zhu, Y. Li, Y. Iwasawa,
377 Y. Matsuo, Z. Xu, and Z. J. Cui. Open X-Embodiment: Robotic learning datasets and RT-X models.
378 <https://arxiv.org/abs/2310.08864>, 2023.

379 [8] J. H. Yang, D. Sadigh, and C. Finn. Polybot: Training one policy across robots while embracing
380 variability. In *7th Annual Conference on Robot Learning*, 2023. URL [https://openreview.net/](https://openreview.net/forum?id=HEIRj51lcS)
381 [forum?id=HEIRj51lcS](https://openreview.net/forum?id=HEIRj51lcS).

- 382 [9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical
383 image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages
384 248–255, 2009. doi:10.1109/CVPR.2009.5206848.
- 385 [10] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite,
386 N. Nabeshima, S. Presser, and C. Leahy. The Pile: An 800gb dataset of diverse text for language
387 modeling. *arXiv preprint arXiv:2101.00027*, 2020.
- 388 [11] G. Penedo, H. Kydliček, L. von Werra, and T. Wolf. Fineweb, 2024. URL [https://huggingface.
389 co/datasets/HuggingFaceFW/fineweb](https://huggingface.co/datasets/HuggingFaceFW/fineweb).
- 390 [12] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang,
391 M. Liu, X. Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of
392 the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.
- 393 [13] A. Albalak, Y. Elazar, S. M. Xie, S. Longpre, N. Lambert, X. Wang, N. Muennighoff, B. Hou, L. Pan,
394 H. Jeong, C. Raffel, S. Chang, T. Hashimoto, and W. Y. Wang. A survey on data selection for
395 language models, 2024.
- 396 [14] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta,
397 C. Mullis, M. Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation
398 image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.
- 399 [15] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu,
400 and R. Martín-Martín. What matters in learning from offline human demonstrations for robot
401 manipulation. *arXiv preprint arXiv:2108.03298*, 2021.
- 402 [16] J. Gao, A. Xie, T. Xiao, C. Finn, and D. Sadigh. Efficient data collection for robotic manipulation via
403 compositional generalization, 2024.
- 404 [17] S. Belkhale, Y. Cui, and D. Sadigh. Data quality in imitation learning. *Advances in Neural Information
405 Processing Systems*, 36, 2024.
- 406 [18] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction
407 to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial
408 intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.
- 409 [19] K. Gandhi, S. Karamcheti, M. Liao, and D. Sadigh. Eliciting compatible demonstrations for multi-
410 human imitation learning. In *Conference on Robot Learning*, pages 1981–1991. PMLR, 2023.
- 411 [20] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu,
412 J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and
413 S. Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and
414 Systems*, Delft, Netherlands, 2024.
- 415 [21] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu.
416 Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and
417 Systems (RSS)*, 2024.
- 418 [22] S. M. Xie, H. Pham, X. Dong, N. Du, H. Liu, Y. Lu, P. S. Liang, Q. V. Le, T. Ma, and
419 A. W. Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. In
420 A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances
421 in Neural Information Processing Systems*, volume 36, pages 69798–69818. Curran Asso-
422 ciates, Inc., 2023. URL [https://proceedings.neurips.cc/paper_files/paper/2023/
423 file/dcba6be91359358c2355cd920da3fcbd-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2023/file/dcba6be91359358c2355cd920da3fcbd-Paper-Conference.pdf).
- 424 [23] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang. Distributionally robust neural networks
425 for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint
426 arXiv:1911.08731*, 2019.

- 427 [24] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor
428 policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.
- 429 [25] M. Reuss, M. Li, X. Jia, and R. Lioutikov. Goal conditioned imitation learning using score-based
430 diffusion policies. In *Robotics: Science and Systems*, 2023.
- 431 [26] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard. Calvin: A benchmark for language-
432 conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation
433 Letters (RA-L)*, 7(3):7327–7334, 2022.
- 434 [27] H. Ha, P. Florence, and S. Song. Scaling up and distilling down: Language-guided robot skill
435 acquisition. In *Proceedings of the 2023 Conference on Robot Learning*, 2023.
- 436 [28] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. Bc-z:
437 Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*,
438 pages 991–1002. PMLR, 2022.
- 439 [29] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn.
440 Robonet: Large-scale multi-robot learning. *arXiv preprint arXiv:1910.11215*, 2019.
- 441 [30] P. Sharma, L. Mohan, L. Pinto, and A. Gupta. Multiple interactions made easy (mime): Large scale
442 demonstrations data for imitation. In *Conference on robot learning*, pages 906–915. PMLR, 2018.
- 443 [31] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay,
444 et al. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference
445 on Robot Learning*, pages 879–893. PMLR, 2018.
- 446 [32] L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot
447 hours. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 3406–3413.
448 IEEE, 2016.
- 449 [33] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, et al. Rt-2:
450 Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot
451 Learning*, pages 2165–2183. PMLR, 2023.
- 452 [34] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen. Learning hand-eye coordination for
453 robotic grasping with deep learning and large-scale data collection. *The International journal of
454 robotics research*, 37(4-5):421–436, 2018.
- 455 [35] K. Burns, Z. Witzel, J. I. Hamid, T. Yu, C. Finn, and K. Hausman. What makes pre-trained visual
456 representations successful for robust manipulation? *arXiv preprint arXiv:2312.12444*, 2023.
- 457 [36] A. Xie, L. Lee, T. Xiao, and C. Finn. Decomposing the generalization gap in imitation learning for
458 visual robotic manipulation. *arXiv preprint arXiv:2307.03659*, 2023.
- 459 [37] S. Young, D. Gandhi, S. Tulsiani, A. Gupta, P. Abbeel, and L. Pinto. Visual imitation made easy. In
460 J. Kober, F. Ramos, and C. Tomlin, editors, *Proceedings of the 2020 Conference on Robot Learning*,
461 volume 155 of *Proceedings of Machine Learning Research*, pages 1992–2005. PMLR, 16–18 Nov
462 2021. URL <https://proceedings.mlr.press/v155/young21a.html>.
- 463 [38] S. Belkhale, Y. Cui, and D. Sadigh. Hydra: Hybrid robot actions for imitation learning. In *Conference
464 on Robot Learning*, pages 2113–2133. PMLR, 2023.
- 465 [39] M. Laskey, J. Lee, R. Fox, A. Dragan, and K. Goldberg. Dart: Noise injection for robust imitation
466 learning. In *Conference on robot learning*, pages 143–156. PMLR, 2017.
- 467 [40] B. Baker, I. Akkaya, P. Zhokov, J. Huizinga, J. Tang, A. Ecoffet, B. Houghton, R. Sampedro, and
468 J. Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in
469 Neural Information Processing Systems*, 35:24639–24654, 2022.

- 470 [41] M. Du, S. Nair, D. Sadigh, and C. Finn. Behavior retrieval: Few-shot imitation learning by querying
471 unlabeled datasets. *arXiv preprint arXiv:2304.08742*, 2023.
- 472 [42] S. Nasiriany, T. Gao, A. Mandlekar, and Y. Zhu. Learning and retrieval from prior data for skill-based
473 imitation learning. In *Conference on Robot Learning (CoRL)*, 2022.
- 474 [43] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- 475 [44] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza,
476 F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv*
477 *preprint arXiv:2304.07193*, 2023.
- 478 [45] H. Xu, S. Xie, X. Tan, P.-Y. Huang, R. Howes, V. Sharma, S.-W. Li, G. Ghosh, L. Zettlemoyer, and
479 C. Feichtenhofer. Demystifying CLIP data. In *The Twelfth International Conference on Learning*
480 *Representations*, 2024. URL <https://openreview.net/forum?id=5BCFlnfE1g>.
- 481 [46] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English,
482 V. Voleti, A. Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets.
483 *arXiv preprint arXiv:2311.15127*, 2023.
- 484 [47] H. V. Vo, V. Khalidov, T. Darcet, T. Moutakanni, N. Smetanin, M. Szafraniec, H. Touvron, C. Couprie,
485 M. Oquab, A. Joulin, et al. Automatic data curation for self-supervised learning: A clustering-based
486 approach. *arXiv preprint arXiv:2405.15613*, 2024.
- 487 [48] K. Chitta, J. M. Álvarez, E. Haussmann, and C. Farabet. Training data subset search with ensemble
488 active learning. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14741–14752, 2021.
- 489 [49] O. Sener and S. Savarese. Active learning for convolutional neural networks: A core-set approach.
490 *arXiv preprint arXiv:1708.00489*, 2017.
- 491 [50] B. Mirzasoleiman, J. Bilmes, and J. Leskovec. Coresets for data-efficient training of machine learning
492 models. In *International Conference on Machine Learning*, pages 6950–6960. PMLR, 2020.
- 493 [51] V. Birodkar, H. Mobahi, and S. Bengio. Semantic redundancies in image-classification datasets: The
494 10% you don’t need. *arXiv preprint arXiv:1901.11409*, 2019.
- 495 [52] K. Tirumala, D. Simig, A. Aghajanyan, and A. Morcos. D4: Improving llm pre-
496 training via document de-duplication and diversification. In A. Oh, T. Naumann,
497 A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural*
498 *Information Processing Systems*, volume 36, pages 53983–53995. Curran Associates,
499 Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a8f8cbd7f7a5fb2c837e578c75e5b615-Paper-Datasets_and_Benchmarks.pdf.
- 501 [53] A. Abbas, K. Tirumala, D. Simig, S. Ganguli, and A. S. Morcos. Semdedup: Data-efficient learning
502 at web-scale through semantic deduplication, 2023.
- 503 [54] S. Mindermann, J. M. Brauner, M. T. Razzak, M. Sharma, A. Kirsch, W. Xu, B. Höltingen, A. N.
504 Gomez, A. Morisot, S. Farquhar, and Y. Gal. Prioritized training on points that are learnable, worth
505 learning, and not yet learnt. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and
506 S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume
507 162 of *Proceedings of Machine Learning Research*, pages 15630–15649. PMLR, 17–23 Jul 2022.
508 URL <https://proceedings.mlr.press/v162/mindermann22a.html>.
- 509 [55] A. H. Jiang, D. L.-K. Wong, G. Zhou, D. G. Andersen, J. Dean, G. R. Ganger, G. Joshi, M. Kaminsky,
510 M. Kozuch, Z. C. Lipton, et al. Accelerating deep learning by focusing on the biggest losers. *arXiv*
511 *preprint arXiv:1910.00762*, 2019.
- 512 [56] M. Paul, S. Ganguli, and G. K. Dziugaite. Deep learning on a data diet: Finding important examples
513 early in training. *Advances in Neural Information Processing Systems*, 34:20596–20607, 2021.

- 514 [57] Y. Oren, S. Sagawa, T. B. Hashimoto, and P. Liang. Distributionally robust language modeling. In
515 *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and*
516 *the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages
517 4227–4237, 2019.
- 518 [58] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan. Theoretically principled trade-off
519 between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482.
520 PMLR, 2019.
- 521 [59] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with
522 accuracy. *arXiv preprint arXiv:1805.12152*, 2018.
- 523 [60] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with
524 low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.
- 525 [61] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine.
526 Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint*
527 *arXiv:2109.13396*, 2021.
- 528 [62] X. Li, K. Hsu, J. Gu, K. Pertsch, O. Mees, H. R. Walke, C. Fu, I. Lunawat, I. Sieh, S. Kirmani,
529 S. Levine, J. Wu, C. Finn, H. Su, Q. Vuong, and T. Xiao. Evaluating real-world robot manipulation
530 policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.
- 531 [63] Z. Fu, T. Z. Zhao, and C. Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost
532 whole-body teleoperation. In *arXiv*, 2024.
- 533 [64] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of*
534 *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- 535 [65] P. Hansen-Estruch, I. Kostrikov, M. Janner, J. G. Kuba, and S. Levine. Idql: Implicit q-learning as an
536 actor-critic method with diffusion policies, 2023.
- 537 [66] B. Sorscher, R. Geirhos, S. Shekhar, S. Ganguli, and A. Morcos. Beyond neural scaling laws: beating
538 power law scaling via data pruning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho,
539 and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 19523–
540 19536. Curran Associates, Inc., 2022. URL [https://proceedings.neurips.cc/paper_files/
541 paper/2022/file/7b75da9b61eda40fa35453ee5d077df6-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/7b75da9b61eda40fa35453ee5d077df6-Paper-Conference.pdf).
- 542 [67] J. Luo, C. Xu, X. Geng, G. Feng, K. Fang, L. Tan, S. Schaal, and S. Levine. Multi-stage cable routing
543 through hierarchical imitation learning. *arXiv pre-print*, 2023. URL [https://arxiv.org/abs/
544 2307.08927](https://arxiv.org/abs/2307.08927).
- 545 [68] E. Rosete-Beas, O. Mees, G. Kalweit, J. Boedecker, and W. Burgard. Latent plans for task agnostic
546 offline reinforcement learning. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*,
547 2022.
- 548 [69] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan,
549 V. Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In
550 *Conference on robot learning*, pages 651–673. PMLR, 2018.
- 551 [70] S. Dass, J. Yapeter, J. Zhang, J. Zhang, K. Pertsch, S. Nikolaidis, and J. J. Lim. Clvr jaco play dataset,
552 2023. URL <https://github.com/clvr-ai/clvr-jaco-play-dataset>.
- 553 [71] G. Zhou, V. Dean, M. K. Srirama, A. Rajeswaran, J. Pari, K. Hatch, A. Jain, T. Yu, P. Abbeel,
554 L. Pinto, et al. Train offline, test online: A real robot learning benchmark. In *2023 IEEE International*
555 *Conference on Robotics and Automation (ICRA)*, pages 9197–9203. IEEE, 2023.
- 556 [72] TensorFlow Datasets, a collection of ready-to-use datasets. [https://www.tensorflow.org/
557 datasets](https://www.tensorflow.org/datasets).

- 558 [73] J. Pari, N. M. Shafiullah, S. P. Arunachalam, and L. Pinto. The surprising effectiveness of representa-
559 tion learning for visual imitation, 2021.
- 560 [74] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural*
561 *information processing systems*, 33:6840–6851, 2020.

562 A Additional Results

563 Here we include results for 10% subsetting of the bridge dataset as described in Section 4.3. In the
564 supplemental material we include videos of rollouts from our experiments.

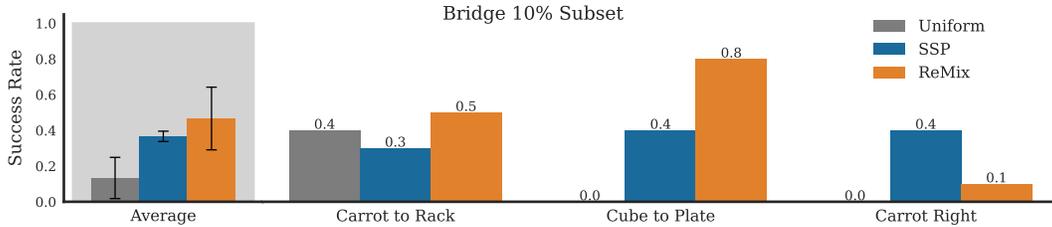


Figure 5: Bridge 10% subsetting.

565 Below we can see the difference in distribution between the BridgeV2 dataset [4] and the Toto dataset
566 [71] in log scale. The Bridge action distribution is far more normal and symmetric than the ToTo action
567 distribution. The Toto distribution is heavily multi-modal and skew.

568 B Dataset Details

569 B.1 OpenX RTX Subset

570 We use a subset of the OpenX Embodiment dataset similar to that used to train the RT-X models [7]. First, we
571 use the RLDS dataset modification repository (https://github.com/kpertsch/rlds_dataset_mod)
572 used by Octo Model Team et al. [20] to preprocess the raw datasets downloaded from Tensor Flow
573 Datasets [72]. Specifically, we resize all images to 256×256 , and filter the Kuka dataset [69] by an
574 included success key. Note that this does warp images. We use the updated version of the Bridge dataset,
575 available at https://rail.eecs.berkeley.edu/datasets/bridge_release/data/tfds/. The
576 specific composition of the dataset is listed in Table 2. Note that we only train on the primary third-person
577 camera in each dataset. For this reason, we omit the NYU Reacher-grabber dataset [73] which *only* includes
578 wrist cameras. We align all action spaces by converting them to delta cartesian and delta euller angle and
579 binarize all gripper actions.

580 B.2 Bridge V2 Dataset

581 For experiments on bridge-only, we split the bridge dataset into 32 domains. First, we re-downloaded
582 the raw bridge dataset and converted it to RLDS using the DLimp convertor (<https://github.com/kvablack/dlimp/>). We then partitioned the bridge dataset by domain using the file path metadata field
583 that lists which setting demonstrations were collected in e.g. “toy-kitchen 1” or “toy-sink-3”. We then
584 manually group the domains into 32 categories. We omitted data that was collected by a scripted policy, as
585 it did not contain the scene information in the filepath metadata. This means we ended up with around
586 45,000 training trajectories, instead of the 60K used in the full bridge dataset. In Table 3 we list the
587 natural weights of each of these domains and the learned weights by Re-Mix. We can see that Re-Mix
588 down-weights some of the largest domains and places their weight on smaller domains.
589

590 B.3 Co-Training Datasets.

591 Below we describe our co-training data and evaluation procedure for the real-world tasks on the WidowX
592 250 and Franka Panda robots.

593 **WidowX Tasks** We evaluate on a 6-DoF WidowX 250 robot on several new pick place tasks in a toy
594 kitchen setting. Our setup is similar to Bridge V2 [4] with a fixed side camera and a blocking controller.
595 Following Walke et al. [4] we use a blocking controller during evaluation. We collect teleoperated
596 demonstrations using an Oculus Quest Headset for motion tracking and co-train on 25 demonstrations for
597 each of the three tasks “Move Cube out of Sink”, “Move Cup into Sink”, and “Move Fork from Sink to
598 Rack.”
599

Domain	Uniform Weight	ReMix Weight
0 toykitchen2	0.18728751	0.0961817
1 datacol2_tabletop_dark_wood	0.094527	0.04846529
2 toykitchen1	0.069307	0.07683
3 toykitchen6	0.06940527	0.0573625
4 datacol2_toykitchen7	0.07133783	0.06905
5 datacol2_toykitchen2	0.0432927	0.03651583
6 toykitchen7	0.032803	0.03538789
7datacol2_folding_table	0.038522	0.0809778049
8 datacol1_toykitchen6	0.03606622	0.037404168
9 datacol2_robot_desk	0.025810027	0.034152
10 datacol2_toykitchen6	0.02394393	0.02740302
11 deepthought_folding_table	0.0272809	0.013906823
12 datacol2_laundry_machine laundry_machine	0.02582954	0.0396389
13 datacol2_toykitchen5, toykitchen5	0.0337366	0.049943
14 deepthought_toykitchen2	0.0253313	0.013434348
15 deepthought_robot_desk	0.01978364	0.032410502
16 tabletop_dark_wood	0.0219985	0.024691
17 datacol2_toysink2 toysink2_bww	0.0225748	0.0198516
18 toykitchen2_room8052	0.01083554	0.0295857
19 deepthought_toykitchen1, datacol1_toykitchen1	0.01868	0.04047
20 datacol2_foldtable_tray, minsky_foldtable_tray, datacol2_toykitchen7_tray	0.037856699	0.0484
21 toysink3_bww, toysink3	0.01235829	0.014877
22 datacol2_toykitchen1	0.01155453	0.02194
23 toysink1_room8052 toysink1	0.00979455	0.01831014
24 tool_chest	0.00471524	0.00878
25 toysink5	0.00405418	2.78E-05
26 whiteboard	0.006774	0.0129337
27 toykitchen4	0.00371938	0.00537445
28 toysink4	0.00289793	1.80E-05
29 toykitchen3	0.00124406	2.72E-05
30 realkitchen1_dishwasher	0.00202648	0.000541
31 tabletop_light_wood, tabletop_white, realkitchen1_counter	0.004647549	0.005079152

Table 3: Learned weights by Re-Mix on the Bridge V2 dataset.

600 During evaluation, we examine generalization on various axes. The “Carrot to Rack” task tests generaliza-
601 tion to picking up a new type of target object, “Cube to Plate” and “Cube to Cup” test generalization to
602 new containers, and “Carrot to Right” tests generalization to both a new target object and a new motion.
603 For each of these tasks, we first take a goal image and then evaluate our policies with fixed object locations
604 for up to 100 seconds, stopping early if the robot or objects reach unrecoverable states. For “Carrot to
605 Rack” we do five trials with the carrot facing down and five trials with it facing upwards. For “Fork to
606 Rack” we use an unseen initial position to the right side of the sink and rotate the fork left 45 degrees for
607 five episodes and to the right 45 degrees for the other five.

608 B.4 Franka Tasks

609 We evaluate on a Franka Panda robot on several pick place tasks on a tabletop. We use a fixed over the
610 shoulder camera We co-train on 25 teleoperated demonstrations for each of the tasks “Pen into Cup,” where
611 we put a pen into a cup from 5 different start locations, and “Flip Bowl,” where a bowl is flipped into a
612 drying rack. For the “Pen into Cup” task we use a different pen than in co-training. However, because our
613 franka embodiment with the Robotiq 2F-85 is not found in our pre-training datasets, we evaluate the same
614 tasks as we co-trained on. We evaluate each start location of the pen twice from a new set of predefined
615 positions. As in the WidowX evaluations, we take a goal image for each task and evaluate for up to 100
616 seconds using a 10Hz controller without blocking control.

	RTX	Bridge
Batch Size	512	384
Action Chunk	4	2
Image Resolution	224×224	224×288

Table 4: Hyperparameters

617 C Training Details

618 **Architecture.** We borrow our architecture from [4] with a few minor changes. Our policies takes as
 619 input a history of two consecutive frames and a single goal image and output a sequence of actions via
 620 DDPM [74].

621 First, we preprocess all images to fit between -1 and 1. Then, we channel-wise concatenate both the goal
 622 image and a grid containing the position of each pixel in (x,y) space also normalized between -1 and 1.
 623 Images are then fed to a ResNet 50 encoder, which employs global average pooling on the output to obtain
 624 a 512 dimension representation for each image. Both image representations are then concatenated and fed
 625 to a diffusion action prediction head.

626 **Hyperparameters.** We use a cosine decay learning rate schedule with an initial learning rate of 0.0002.
 627 We train all models for 400K steps and evaluate the final checkpoint, except for Bridge 10% subsetting,
 628 which we found to perform better after 200K steps. More detailed hyperparameters are found in Table 4.
 629 Note that there are some differences between bridge and RTX which were made for computational reasons
 630 – we iterated faster on the bridge dataset before scaling to RTX. We also did maintained aspect ratio for
 631 bridge, hence the different image input size, but did not for RTX follow Octo Model Team et al. [20]. We
 632 apply data augmentation to all images consistently across the time horizon and goal image (meaning that
 633 the goal image and all past images of each example have the same augmentation applied). We use random
 634 resize cropping, brightness, contrast, and hue randomization. For k-means in SSP for Bridge we set $k = 32$,
 635 equal to the number of domains used for Re-Mix.

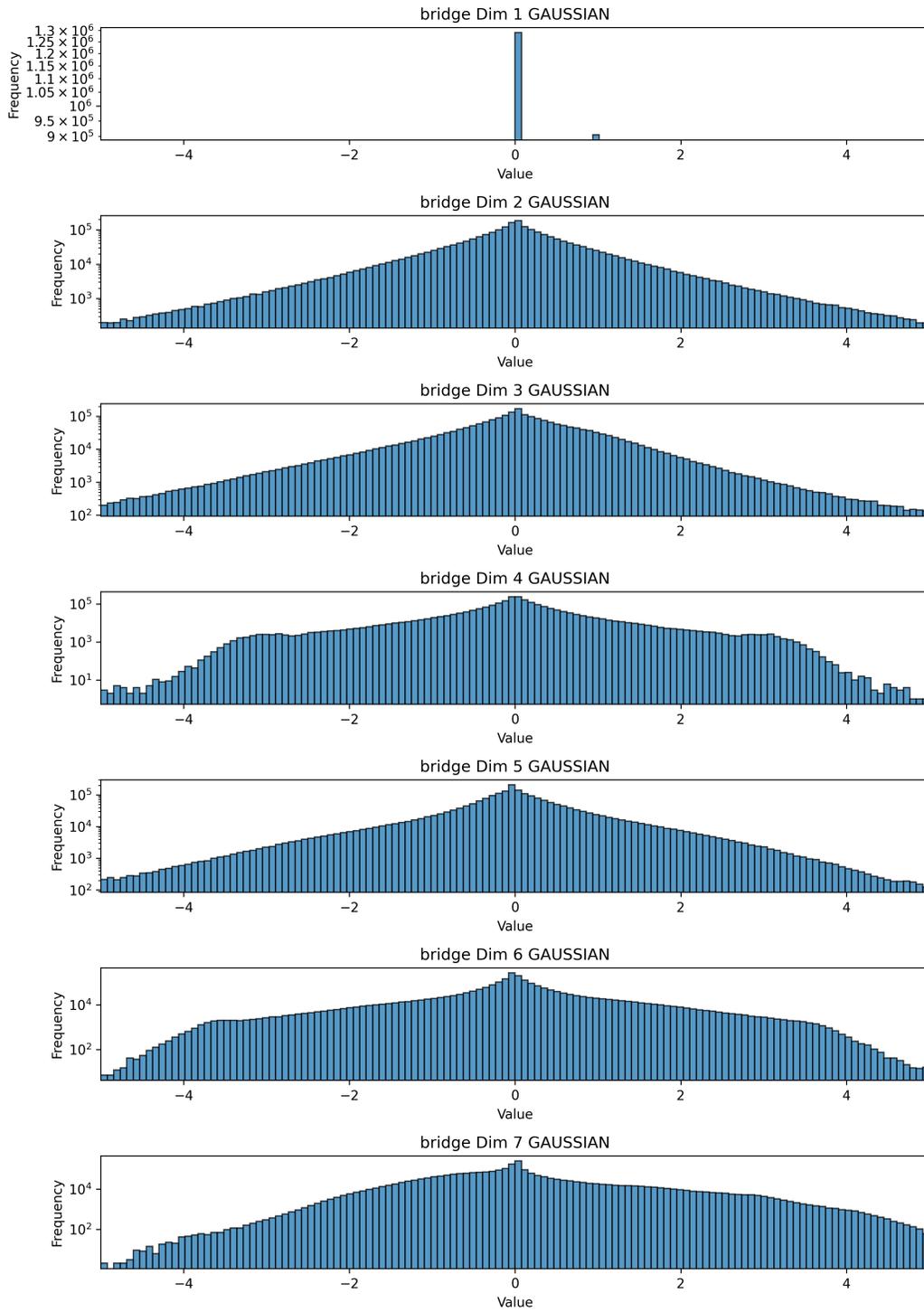


Figure 6: Action distributions for Bridge.

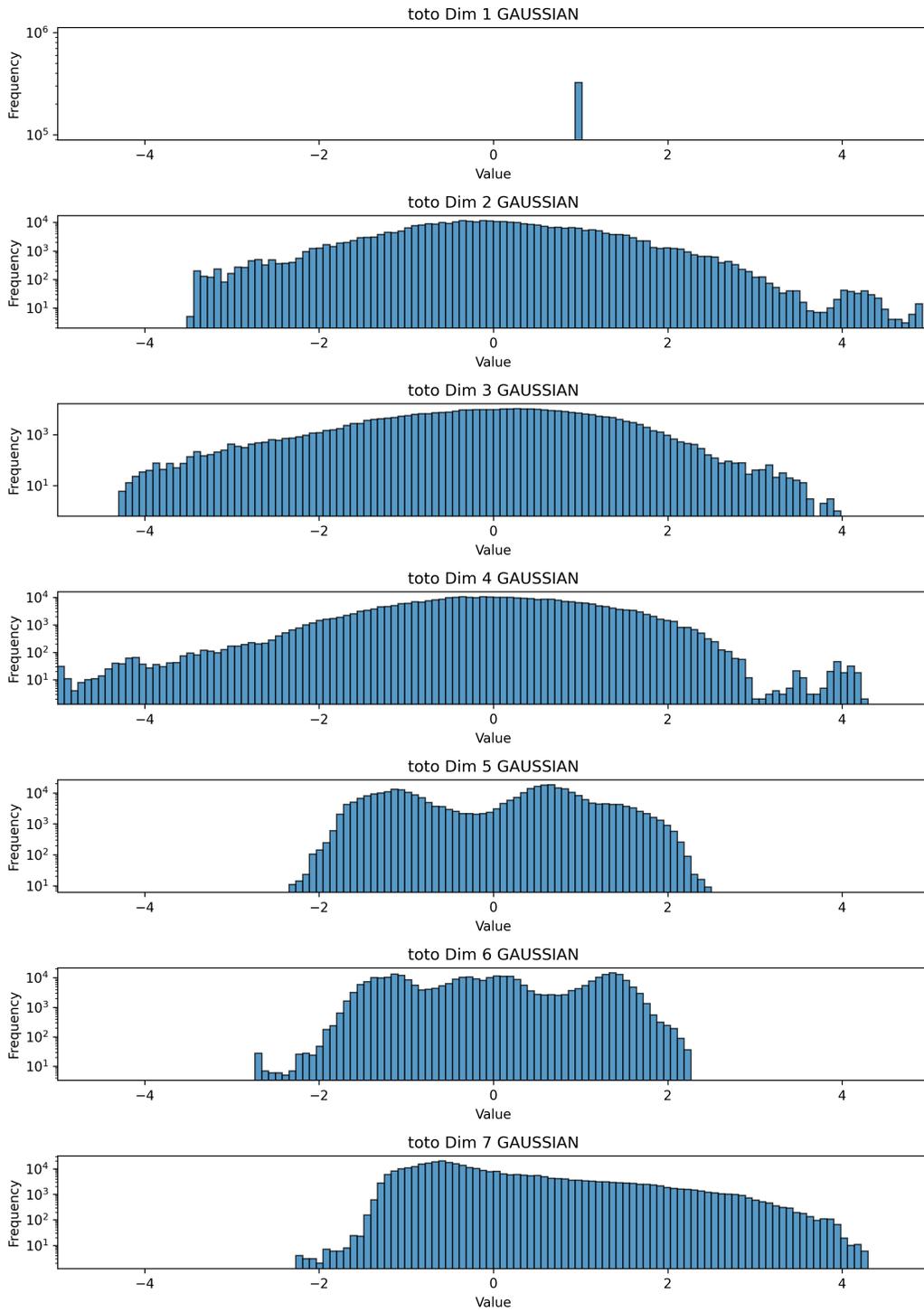


Figure 7: Action distributions for Toto.