

## A Additional Results

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

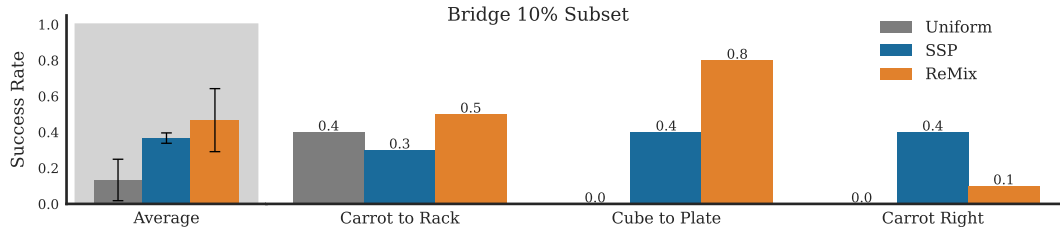


Figure 5: Bridge 10% subsetting.

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

## B Dataset Details

### B.1 OpenX RTX Subset

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

### B.2 Bridge V2 Dataset

For experiments on bridge-only, we split the bridge dataset into 32 domains. First, we re-downloaded 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 that lists which setting demonstrations were collected in e.g. “toy-kitchen 1” or “toy-sink-3”. We then manually group the domains into 32 categories. We omitted data that was collected by a scripted policy, as it did not contain the scene information in the filepath metadata. This means we ended up with around 45,000 training trajectories, instead of the 60K used in the full bridge dataset. In Table 3 we list the natural weights of each of these domains and the learned weights by Re-Mix. We can see that Re-Mix down-weights some of the largest domains and places their weight on smaller domains.

### B.3 Co-Training Datasets.

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

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

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
7 datacol2_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 deeptthought_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 deeptthought_toykitchen2	0.0253313	0.013434348
15 deeptthought_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 deeptthought_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.

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

#### B.4 Franka Tasks

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

	RTX	Bridge
Batch Size	512	384
Action Chunk	4	2
Image Resolution	$224 \times 224$	$224 \times 288$

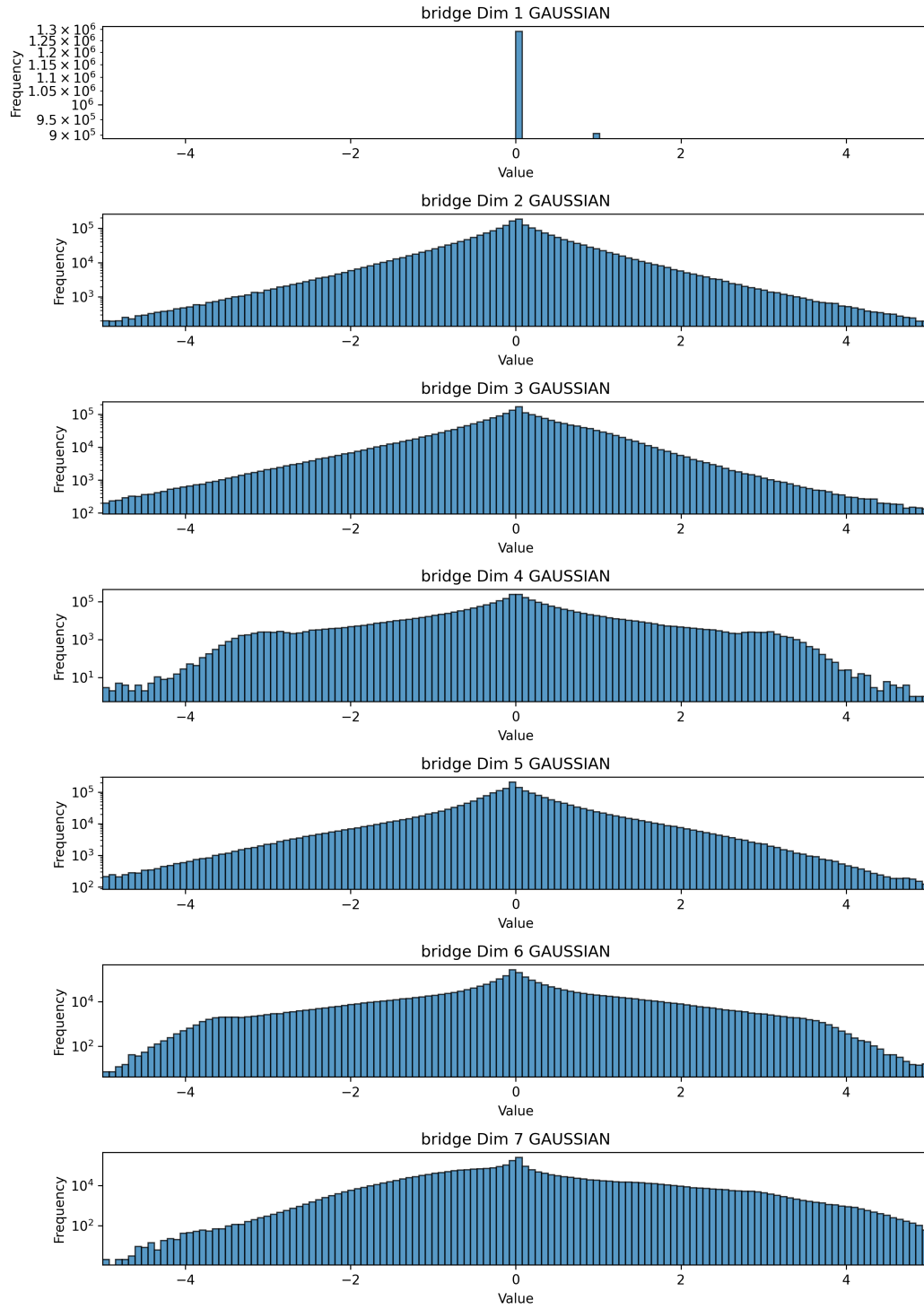
**Table 4:** Hyperparameters

## 617 C Training Details

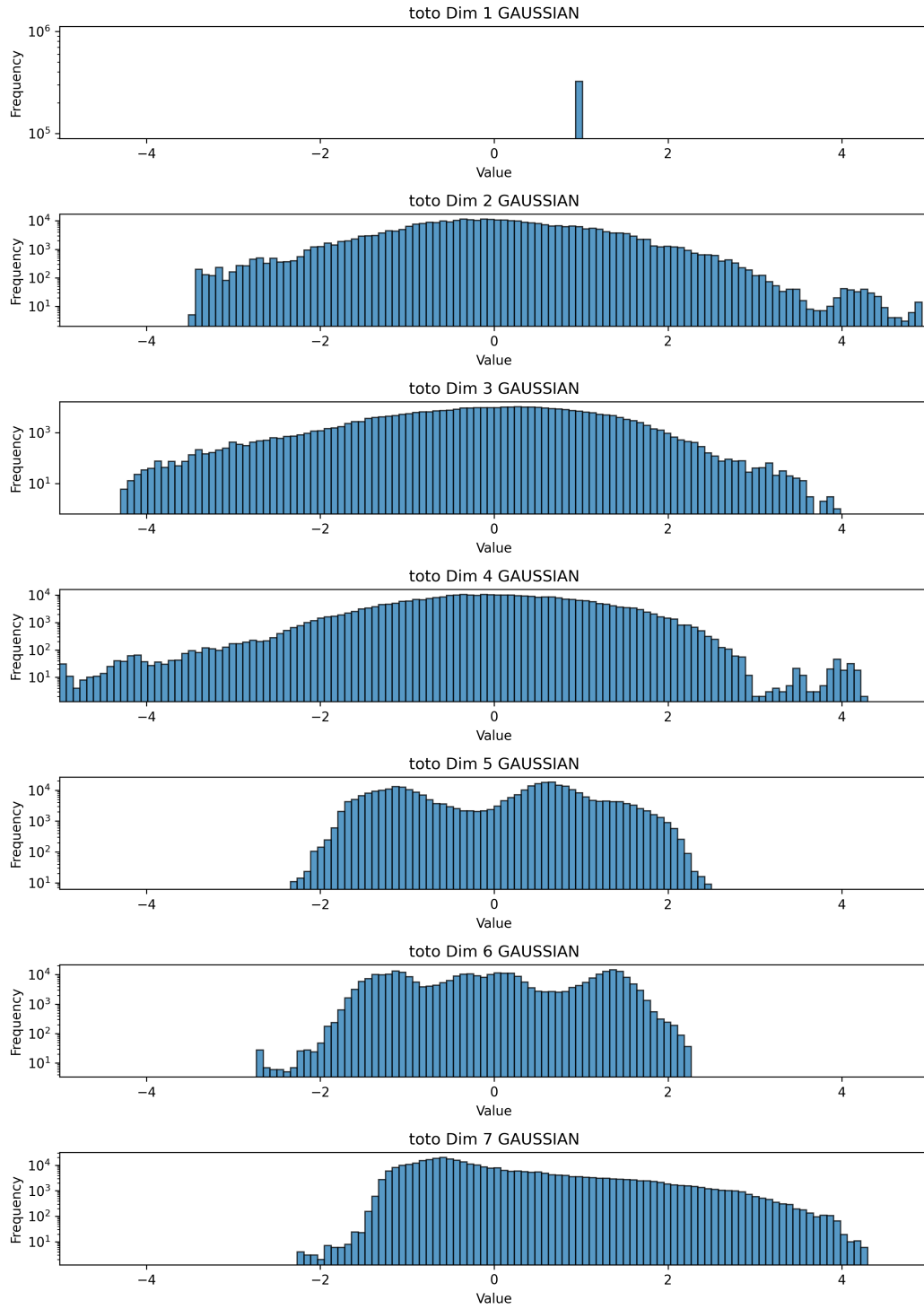
618 **Architecture.** We borrow our architecture from [4] with a few minor changes. Our policies takes as  
619 input a history of two consecutive frames and a single goal image and output a sequence of actions via  
620 DDPM [74].

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.