

Stabilize to Act: Learning to Coordinate for Bimanual Manipulation Supplementary Material

A Training Details

We provide details for training each of the models for BUDS: f_{θ}^k and f_{ψ}^r for the stabilizing policy and π_{ϕ}^a and f^g for the acting policy.

A.1 Stabilizing Policy Training

The keypoint models f_{θ}^k is trained with a hand-labelled dataset of 30 pairs of images and human-annotated keypoints. We augment each image 10X with a series of label-preserving transformations from ImgAug library [39], including rotation, blurring, hue and saturation changes, affine transformations, and adding Gaussian Noise. The detailed parameters for the transformations are listed in Table 3 and we visualize the image augmentations in Fig. 5. The restabilizing classifier f_{ψ}^r is trained on a dataset of images from 20 demonstration rollouts with 100 images each. Each image is paired with binary expert annotation of whether or not restabilizing is needed and augmented by 2X with the same image transformations from above.

Both the keypoint model and the restabilizing classifier are trained against a binary cross-entropy loss with an Adam [41] optimizer. The learning rate is $1.0e^{-4}$ and the weight decay is $1.0e^{-4}$ during the training process. We train these models for 25 epochs on a NVIDIA GeForce GTX 1070 GPU for 1 hour.

A.2 Acting Policy Training

The acting policy starts from a pre-grasped position, which we achieve using an optional grasping keypoint model. The training procedure of grasping keypoint model f^g is the same as that of stabilizing keypoint model f_{θ}^k . After the robotic gripper grasps the object, we collect 20 acting demonstration rollouts, each with between 50 and 200 steps. The variation of 20 demonstrations comes from the randomization of initial object position, differences in object shape and dynamics, and variations in grasps. With these demonstrations, we use one set of hyperparameters for all tasks to train a BC-RNN model similar to prior work [42]. We load batches of size 100 with a history length of 20. We learn policies from input images and use a ResNet-18 [38] architecture encoder which is trained end-to-end. These image encodings are of size 64 and are then concatenated to the proprioceptive input p_t to be passed into the recurrent neural network which uses a hidden size of 1000. We train against the standard imitation learning loss with a learning rate of

Augmentation	Parameters
LinearContrast	(0.95, 1.05)
Add	(-10, 10)
GammaContrast	(0.95, 1.05)
GaussianBlur	(0.0, 0.6)
MultiplySaturation	(0.95, 1.05)
AdditiveGaussianNoise	(0, 3.1875)
Scale	(1.0, 1.2)
Translate Percent	(-0.08, 0.08)
Rotate	(-15°, 15°)
Shear	(-8°, 8°)
Cval	(0, 20)
Mode	['constant', 'edge']

Table 3: **Image Data Augmentation Parameters:** We report the parameters for the data augmentation techniques used to train the stabilizing policy’s stabilizing position and restabilizing classifier models in BUDS. All augmentations are used from the imgaug Python library [39].

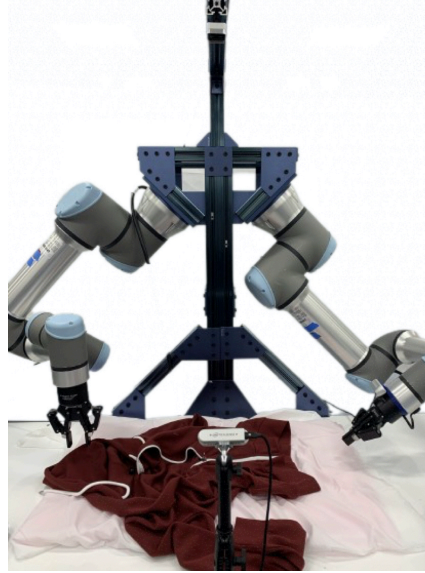


Figure 4: **Experimental Setup:** We present our experimental setup, which uses three cameras due to heavy occlusion during manipulation. One camera is mounted overhead, one is on the wrist of the right arm, and one is facing the front of the workspace at an angle.

501 $1e^{-4}$ and a weight decay of 0. We train for 150k epochs on NVIDIA GeForce GTX 1070 GPU for
502 16 hrs.

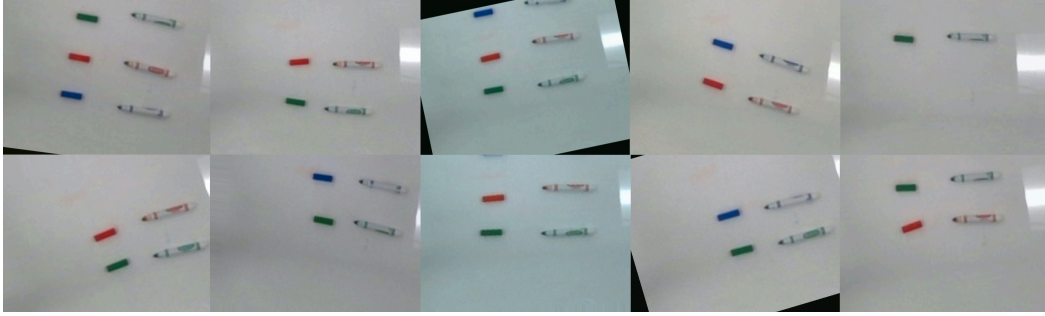


Figure 5: **Data Augmentation for Image Datasets:** We visualize images from the augmented dataset used to train the stabilizing position model and restabilizing classifier for the marker capping task’s stabilizing policy: f_{θ}^k and f_{ψ}^r . For f_{θ}^k , the dataset of expert-labelled image and keypoint annotations is augmented 10X to construct a final dataset of size 300. For f_{ψ}^r , the dataset is augmented 2X for a final size of 4000 image and binary classification pairs.

503 B Experiment Details

504 For all tasks, BUDS’s acting policy uses a 3D action space. For the three tasks other than Pepper
505 Grinder, this action space represents change in end effector position, $(\Delta x, \Delta y, \Delta z)$. For the Pepper
506 Grinder task, this action space instead represents the change in end effector roll, pitch, and yaw,
507 due to safety concerns involving the closed chain constraint created by using both arms to grasp the
508 pepper grinder tool.

509 All tasks use the overhead camera for the sta-
510 bilizing keypoint model and grasping model in-
511 puts. Depending on the task and the types of
512 occlusion present during manipulation, we use
513 two of the three cameras for the acting policy
514 and the restabilizing classifier as outlined in Ta-
515 ble 4.

516 We use the optional grasping model f^g for all
517 tasks except the Pepper Grinder task to ac-
518 count for variations of the initial positions of
519 the jacket, markers, and vegetables. Instead for
520 the Pepper Grinder task, the acting arm instead
521 moves to the point corresponding to the end effector position of the stabilizing arm, and grasps at
522 a fixed height above the stabilizing arm corresponding to the height of the pepper grinder. The
523 pepper grinder begins pregrasped in the stabilizing robot hand, but the plate positions are randomly
524 initialized.

525 In the **BC-Stabilizer** baseline, the stabilizing policy learned via imitation learning is trained with
526 the same procedure as the acting policy for BUDS, with the exception of using an output of two
527 Gaussian mixtures to cover the 3D $(\Delta x, \Delta y, \Delta z)$ action space.





Task	Cameras
 Pepper Grinder	Overhead, Side
 Jacket Zip	Overhead, Side
 Marker Cap	Overhead, Wrist
 Cut Vegetable	Wrist, Side

Table 4: **Task-Specific Cameras:** We report the cameras used for obtaining images as input for the acting policy and restabilizing classifier by task.