

6 Limitations

Our results demonstrate that CrossFormer can match the performance of specialist policies trained on only the most relevant data for a given embodiment, but they do not yet show significant positive transfer across embodiments. We anticipate that as we train on larger robot datasets with more embodiments, we will see greater positive transfer. Importantly, CrossFormer does not require any additional engineering to add data from new embodiments with different observation or action types, making scaling the training data straightforward. Another limitation is that our data mix uses hand-picked sampling weights to avoid over-training on datasets with many repetitive episodes and under-training on the data most relevant to our evaluation settings. In principle, as we scale model size, the policy should have the capacity to fit all the data equally well without any data weighting. Finally, given that we need large models to fit large multi-robot datasets, the model’s inference speed can become a limiting factor. In this work we successfully applied our policy to a high-frequency, fine-grained bimanual manipulation task, but as we scale the model’s size we may not be able to control these higher frequency embodiments. Future hardware improvements will help to alleviate the issue, but further research is needed on techniques for using large models to control high-frequency robots.

More information and videos can be found at our anonymized website: crossformer.github.io.

A Training Data

We list the sampling weights for each dataset in our data mixture in Table 1. We up-weight our target datasets: Bridge, ALOHA-multi-task, GNM, Go1-walk, and Franka-wiping.

CrossFormer Dataset Mixture	
Fractal [32]	17%
Kuka [28]	2.2%
BC-Z [57]	2.2%
Stanford Hydra Dataset [58]	0.015%
Language Table [59]	1.5%
Taco Play [60, 61]	1.2%
Furniture Bench Dataset [62]	0.83%
UTAustin Mutex [63]	0.76%
Austin Sailor Dataset [64]	0.74%
Roboturk [65]	0.79%
Toto [66]	0.68%
Austin Sirius Dataset [67]	0.59%
Berkeley Autolab UR5 [68]	0.41%
IAMLab CMU Pickup Insert [69]	0.31%
Viola [70]	0.32%
Berkeley Fanuc Manipulation [71]	0.26%
NYU Franka Play Dataset [72]	0.28%
Jaco Play [73]	1.6%
Berkeley Cable Routing [74]	0.089%
Austin Buds Dataset [75]	0.072%
CMU Stretch [76]	0.053%
DLR EDAN Shared Control [77]	0.019%
DROID [35]	0.022%
Bridge [37, 36]	17%
GNM [41]	17%
ALOHA-multi-task	17%
Go1-walk	8.5%
Franka-wiping	8.5%

Table 1: The CrossFormer training data mixture uses datasets from the Open X-Embodiment dataset [5] and additional data collected for this project.

Hyperparameter	Value
Optimizer	AdamW [52]
Learning Rate	3e-4
Warmup Steps	2000
LR Scheduler	reciprocal square-root
Weight Decay	0.1
Gradient Clip Threshold	1
Batch Size	1024
Layers	12
Attention heads	8
Token embedding size	512
MLP dimension	2048
Context length	1890
Total training steps	300K

Table 2: Training hyperparameters for CrossFormer.

B Training Hyperparameters

In Table 2 we list the hyperparameters for the optimizer and policy architecture. In total, along with the ResNet-26 image encoders and action heads, our model has 110M parameters. We initialize the ResNet-26 encoders with ImageNet pre-trained weights. Training took 80 hours on a TPU V5e-256 pod. We apply color jitter and random resizing/cropping image augmentations. During training, we use hindsight goal relabeling and sample future observations uniformly at random to use as goals [54]. If a language instruction is available for a trajectory, we randomly mask either the language or goal so that at test time we can condition our policy using either task specification [55].

C Evaluation Setups

Below we provide further details on our evaluation settings (see Fig. 4 for images):

WidowX Manipulation We use the Bridge setup from Walke et al. [36]. We use an over-the-shoulder camera view and sample actions from the single arm head of our policy. We evaluate for 12 trials on the language-conditioned task of putting a spoon on a cloth and 12 trials on the goal-conditioned task of putting a mushroom in a pot. Positions of the spoon, cloth, mushroom, and pot are varied between trials.

Franka Manipulation We use the DROID setup from Khazatsky et al. [35]. We use an over-the-shoulder camera view and sample actions from the single arm head of our policy. We evaluate for nine trials on the language-conditioned task of using a sponge to sweep pinecones into a dustpan. The position of the sponge, pinecones, and dustpan are varied between trials.

ALOHA Bimanual Manipulation We use the ALOHA setup from Zhao et al. [46]. We use three camera views, one overhead and two wrist, and sample actions from the single arm head of our policy. We perform 10 trials over one language-conditioned task of taking the cap off of a pen. The position of the pen is varied between trials.

LoCoBot Navigation We use the LoCoBot setup from Shah et al. [42] which has one camera view. We evaluate on suite of three skills: path-following, obstacle avoidance, and sharp corner-turning. We sample actions from the navigation head of our policy. We combine our policy with the graph-based planner and distance function from Shah et al. [42]. We first obtain a topological map \mathcal{M} of the environment by teleoperating the robot. Then, at every timestep we find the closest node in \mathcal{M} , search the graph for the shortest path from this node to the goal, and command the policy with the most immediate subgoal in the path. We evaluate the success of a trajectory based on the number of subgoals between the closest node at the end of the trajectory and the goal.

606 **Go1 Quadruped** We evaluate on a Unitree Go1 which uses proprioceptive observations $o_t \in$
607 \mathcal{R}^{59} . We sample actions from the quadruped head of our policy, and the task is to walk forward.
608 Importantly, unlike prior work [8], we directly control the quadruped’s joints rather than doing
609 higher level control with navigation waypoints. As our evaluation metric we report the percentage
610 of the reward achieved by the RL-trained expert policy that generated the data (see Section 3.1).

611 **Tello Quadcopter** Finally, we perform evaluation on a Tello quadcopter using the navigation head
612 of our policy. Since the navigation head outputs 2-D relative waypoints, we maintain a static height
613 throughout the trajectory [42, 41]. Notably, we do not train on quadcopter data so this setting
614 requires *zero-shot* generalization to a new embodiment.