EC-Flow: Enabling Versatile Robotic Manipulation from Action-Unlabeled Videos via Embodiment-Centric Flow

Supplementary Material

A. Limitations and Future Work

EC-Flow performs manipulation starting from a manually set initial pose and computes the subsequent actions. Future work may involve integrating the off-the-shelf grasping pose generation models [2, 7] to facilitate starting position initialization, thereby enhancing overall task efficiency.

Additionally, we aim to leverage foundation multi-modal models to extract gripper state information directly from the video dataset, which could then be integrated into the flow prediction network to facilitate gripper state prediction.

B. Design Choices of EC-Flow

B.1. Trade-offs between EC-Flow and End-Effector-Regression-Only Methods

End-effector-regression methods refer to those that directly regress the end-effector (EEF) pose using re-projection error, without optimizing the full joint configuration. Table 1 compares the inference latency and task success rates of different methods. For our EC-Flow approach, flow prediction requires 4.37 s for 8 frames—performed only once at the start of each trajectory and not involved during execution. This step could be further accelerated by replacing diffusion with flow matching. The point projection step takes 0.01 s, while action computation takes 0.21 s when regressing only the EEF pose, or 0.37 s when performing full-joint optimization, due to the added cost of inverse kinematics and optimization complexity.

In terms of task performance, the EEF-only baseline shows a drop in success rates by 5.3% in simulation and 7.1% in the real world, while achieving 1.76× faster inference. The performance drop in simulation is mainly due to the *door-open* task, where the EEF becomes heavily occluded. In the real world, it stems from the *fold-towel* task, where the EEF shifts direction significantly—from the side to the front—making it difficult to track the initial side-view points. To address these challenges, we adopt full-joint optimization, which helps mitigate occlusions and better accommodate large directional changes of the EEF, at a moderate computational cost. Additionally, joint-specific weights in the re-projection error can be manually tuned based on joint visibility or task requirements. To improve optimization convergence and stability, we initialize from the previous pose estimate.

Metric	Flow	3D Points	Action Calc.	
	Pred.	Proj.	EEF Only	Full Joints
Inf. Latency	4.37 s (only once)	0.01 s	0.21 s	0.37 s
Succ. Rate (sim / real)	-	-	66.7% / 70.0%	72.0% / 77.1%

Table 1. Inference latency and success rate of EC-Flow.

B.2. Goal Image VS. Object Flow Prediction

We initially explored object flow to support EC-Flow by modeling object interaction, but found it struggles to converge on deformable objects due to unstructured motion patterns (such as folding towel), while goal image converges faster and could also serve as a proxy for modeling interaction. However, we agree that for complex tasks with multiple required intermediate object states, object flow could be more useful than using goal-image alone.

B.3. Sensitivity to Camera Pose Selection

Our EC-Flow method does not rely on a specific camera pose that captures all robot joints in view. In practice, having visibility of just 2–3 joints is sufficient for reliable action computation. EC-Flow allows flexible inclusion of any visible joints in the optimization, adapting dynamically based on their visibility.

However, it is crucial that the set of visible joints remains consistent across frames. Since EC-flow is predicted only from the initial frame, if a joint visible at the beginning becomes fully occluded later, the system may fail to track it, potentially leading to execution failure.

C. Use of Internet Cross-Embodiment Data

To investigate the potential of cross-embodiment video data under limited robot demonstrations, we conduct a preliminary study on two representative tasks—*door-open* and *door-close*—from the Meta-World benchmark [6]. We augment a small set of robot demonstrations with 50 human videos of the same tasks sourced from the Something-Something-v2 dataset [3].

We evaluate in a low-data regime with only 2 robot demonstrations per task (compared to 5 in the standard setting). As shown in Table 2, human video data alone fails to achieve zero-shot transfer to the robot embodiment. Using 2 robot demos achieves a 46% success rate; adding 5 human videos improves this to 52%, while incorporating 50 human videos further boosts performance to 70%.

These results suggest that human videos provide valuable motion priors and can significantly enhance sample efficiency in robot learning. We believe that large-scale pretraining on internet-scale human video datasets, followed by robot-specific fine-tuning, represents a promising direction for future research.

Data Comp.	50👤	2🤖	5👤+ 2🤖	50👤+ 2🤖
Succ. Rate	0%	46%	52%	70%

Table 2. Success rate for robot and human data compositions.

D. Implementation Details

D.1. Real-World Setup

Our real-world setup is illustrated in Figure 1.

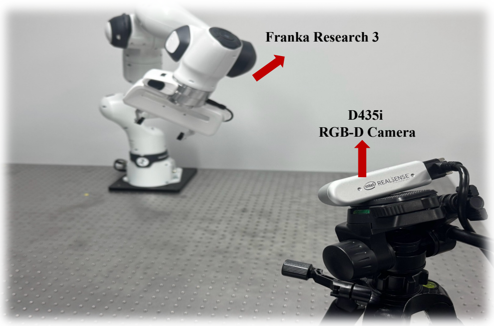

Figure 1. The real-world setup of VERM.

D.2. Real-world Baselines

(1) BC: Standard behavior cloning baseline with dual-stream encoder architecture. Processes RGB observations through ResNet-18 [4] and language instructions via CLIP [5], trained end-to-end on action-labeled demonstrations.

(2) Track-2-Act [1]: An object-centric flow prediction method that predicts object flow and calculates actions from object movements.

D.3. Parameter Details

The parameters of the flow prediction and goal-image prediction network are shown in Table 3 and Table 4 respectively.

E. More Visualization Results

We demonstrate more visualization results of EC-Flow in the Meta-World (Figure 2) and real-world (Figure 3). The demonstration videos can be found on our project website.

Parameter	Value
horizon	8
num_points	400
img_size	128
num_sampling_steps	250
batch_size	56
optimizer	AdamW
lr	5.0e-5
loss_weight	1
transformer depth	24
transformer hidden_size	1152
transformer num_heads	16
mlp_ratio	4.0
lang_dim	1024
img_dim	512

Table 3. Hyper-parameters of flow prediction network.

Parameter	Value
horizon	8
num_points	400
img_size	128
num_sampling_steps	250
batch_size	56
optimizer	AdamW
lr	1.0e-4
loss_weight	0.4
transformer depth	12
transformer hidden_size	384
transformer num_heads	6
patch_size	16
lang_dim	1024
img_dim	512

Table 4. Hyper-parameters of goal image prediction network.

References

[1] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision*, 2024. 3

[2] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11444–11453, 2020. 2

[3] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5842–5850, 2017. 2

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.

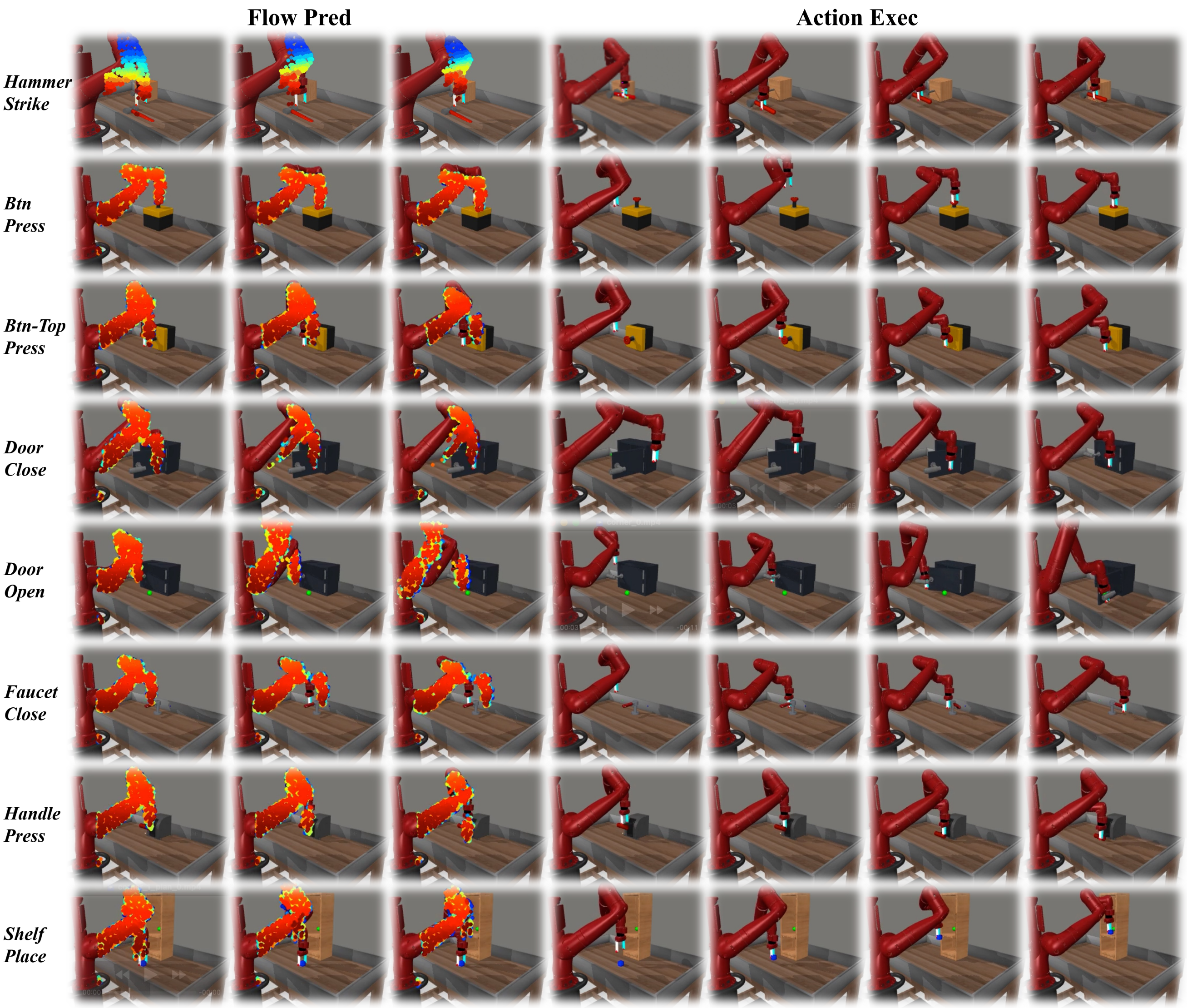

Figure 2. Visualization of the EC-Flow in Meta-World.

Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 3

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 3

[6] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, 2019. 2

[7] XinQiang Yu, Jialiang Zhang, Haoran Liu, Danshi Li, Haoran Geng, He Wang, Yufei Ding, and Jiayi Chen. Dexgraspnet 2.0: Learning generative dexterous grasping in large-scale synthetic cluttered scenes. In *2nd Workshop on Dexterous Manipulation: Design, Perception and Control (RSS)*, 2024. 2

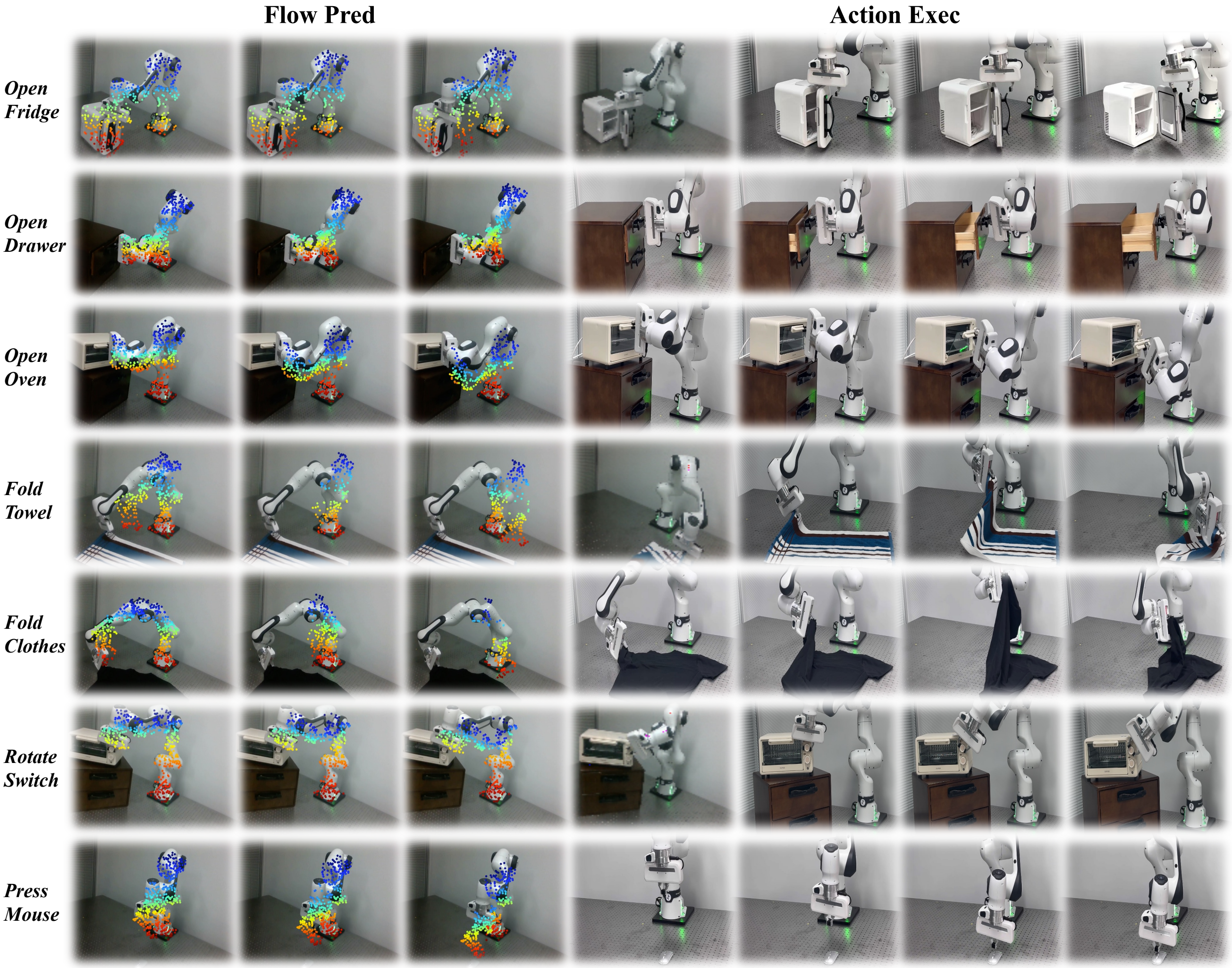

Figure 3. Visualization of the EC-Flow in the real-world.