Feel the Force: Contact-Driven Learning from Humans

Ademi Adeniji*[1,2] Zhuoran Chen*[1] Vincent Liu[1] Venkatesh Pattabiraman[1] Siddhant Haldar[1]
Raunaq Bhirangi[1] Pieter Abbeel[2] Lerrel Pinto[1]
[1]New York University [2]UC Berkeley
*Equal Contribution

***Index Terms*—Learning from Touch, Learning from Humans, Imitation Learning**

I. Introduction

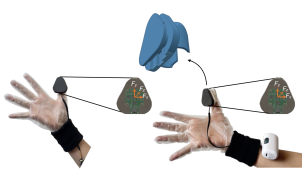
Humans perform fine-grained manipulation by modulating contact forces through both vision and rich tactile sensing. However, such low-level contact reasoning remains a fundamental challenge in robotics due to sparse and delayed sensory feedback [15, 12]. Prior tactile imitation learning approaches have integrated force signals [11], but typically rely on teleoperation [7] or expensive haptic interfaces [9], and require visual cues for continuous control [17]. Although glove-based data collection systems offer scalable alternatives [14], they often lack force sensing and rely on hand-crafted priors. Furthermore, methods that passively input tactile signals into policies [16, 9, 12, 11] struggle to generalize due to embodiment mismatch and require large-scale data to compensate for increased input complexity.

In this work, we ask: Can we endow robots with robust, force-aware control by learning efficiently from human tactile experiences? We present FeelTheForce (FTF), a framework that enables robots to learn force-aware control directly from human tactile demonstrations. FTF models tactile-proprioceptive signals collected via a glove and trains a transformer-based policy to predict hand trajectories and contact forces. These are retargeted to robot end-effector poses, and a low-level PD controller modulates the gripper to track predicted forces—enabling precise control without any robot training data. Unlike teleoperation-based methods, FTF decouples learning from execution and leverages natural human demonstrations for robust manipulation.

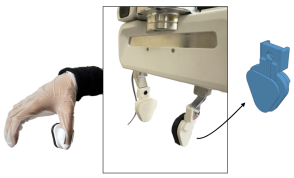
This formulation offers two key advantages: (1) it eliminates the need for large-scale robot data and expensive haptic teleoperation and (2) it enables generalization from the human embodiment to the robot embodiment to solve force-sensitive tasks robustly. We transfer the learned policy to a Franka Panda robot with fingertip tactile sensors and evaluate on 5 force-sensitive manipulation tasks.

In summary, we demonstrate that:

- FTF robustly solves all 5 force-sensitive tasks evaluated with a 77% success rate where baselines fail showing that active force prediction and reproduction is more effective than passive use of multi-modal force inputs.



(a) AnySkin augmented glove, worn by a human data-collector. The straightforward electronics of the sensor interface both reduces excessive wiring and also allows for a bluetooth setting (right).



(b) Middle: Franka Panda gripper with AnySkin on one fingertip, emulating the human wearable (left). We attach a plain silicone cap on the other fingertip.

Fig. 1: Hardware setup for human demonstration and robot replication using AnySkin [11].

- FTF achieves higher success rates than baselines trained on robot teleoperation data showing that the natural data collection enabled by the tactile glove can be effective for tactile data collection.
- FTF is able to achieve a success rate of 67% on a task with adversarial disturbances during deployment, displaying robustness to test-time shifts in the tactile data distribution.

II. FTF

FTF collects tactile data from human demonstrations using a low-cost force-sensing glove and learning policies that predict both actions and desired forces from combined visual and tactile inputs.

Data Acquisition for Human-to-Robot Force Transfer FTF enables task execution through natural human movements. During data acquisition, as the human performs the task, two calibrated RealSense cameras record visual observations of the hand and environment. At deployment, the same camera setup monitors a Franka Panda arm in the same environment. During human data collection, we use a custom tactile glove inspired by AnySkin [1], featuring 3D-printed magnetometer-based sensors placed under the thumb to avoid blocking manipulation. The glove maintains visual transparency and streams 3D force data via USB at 200 Hz. We use the norm of the center sensor's force vector as the

TABLE I: Performance comparison of different gripper action spaces in Human Demo

Task	FTF	Binary Gripper	Continuous Gripper
`Place soft bread on plate`	13/15	0/15	0/15
`Unstack single plastic cup from stack`	9/15	0/15	0/15 (2/15 picked 3 cups)
`Place egg in pot`	13/15	0/15	0/15
`Place bag of chips on plate`	10/15	0/15	0/15
`Twist and lift bottle cap`	13/15	11/15 (1/15 break gripper pads)	0/15

TABLE II: Performance comparison of different gripper action spaces in Robot Teleop Demo

Task	FTF	Binary Gripper	Continuous Gripper
`Place soft bread on plate`	5/15	0/15	3/15
`Unstack single plastic cup from stack`	4/15	0/15 (6/15 picked 3 cups)	0/15 (2/15 picked 2 cups)
`Place egg in pot`	0/15	0/15	0/15
`Place bag of chips on plate`	3/15	0/15	0/15
`Twist and lift bottle cap`	9/15	12/15	8/15

aggregated force signal. A schematic is shown in Figure 1a. The force data is transferred to the robot using gripper-mounted tactile sensors (see Figure 1b).

Embodiment Agnostic Scene Representation We convert human hand motion into a point-based representation using Mediapipe [10] to extract 2D keypoints from two calibrated camera views, which are triangulated into 3D. The robot's position is defined by the midpoint of the thumb and index finger, and orientation is recovered from hand pose changes. This pose is converted into a fixed set of robot keypoints. We also record the gripper state based on finger distance and include tactile force from the glove at each timestep.

For the environment, sparse object keypoints are annotated once and propagated across demonstrations using DIFT [13], then tracked over time with Co-Tracker [6]. These keypoints are triangulated into 3D points. At inference, DIFT initializes the points, and Co-Tracker updates them during execution.

Policy Learning We use a transformer policy [4, 3] that takes as input robot and object keypoints, gripper state, and force value. These inputs are tokenized via MLP encoders and fed into the transformer, which predicts future robot keypoints, gripper actions, and contact forces. To ensure temporal smoothness, we apply action chunking with exponential averaging [17, 2]. The policy is trained with mean squared error on predicted trajectories.

Inference At each timestep, the model predicts 3D robot keypoints, from which we recover the end-effector pose using rigid-body geometry. It also outputs the desired contact force and gripper state. If the gripper should close, a PD controller adjusts the closure until the measured force matches the target. Otherwise, the gripper opens directly. The robot then executes the predicted motion, and object keypoints are updated with Co-Tracker.

III. Experiments

Task Descriptions We evaluate FTF on five real-world force-sensitive manipulation tasks involving fragile and deformable objects, with 30 human and 30 teleoperated demonstrations per task. Our manipulation tasks involve variations designed to evaluate the scope of force-sensitive manipulation capabilities achievable with FTF (details see Appendix V-B).

FTF outperforms baselines on tasks requiring delicate manipulation. As shown in Table I, it is the only method that reliably isolates a single cup during unstacking and avoids over-gripping in bottle cap removal. Binary grippers apply excessive force, while continuous grippers suffer from instability due to imprecise finger-to-gripper mapping.

FTF also outperforms teleoperation baselines that use force data passively. As shown in Table II, it achieves higher success rates than both binary and continuous gripper strategies trained on teleoperated data, demonstrating the benefits of explicit force prediction. However, in tasks like `place egg in pot` and `twist and lift bottle cap`, noisy force signals from teleoperation reduce performance, and the method fails to outperform the binary baseline. Continuous grippers struggle with deformable objects due to sample inefficiency and varying force requirements across executions.

IV. Conclusion and Limitations

We present FTF, a novel framework for learning force-sensitive manipulation from human tactile demonstrations. By leveraging a tactile glove and vision-based hand pose estimation, FTF captures rich contact force signals from natural human interactions without relying on teleoperation or robot-collected data. Our system trains a closed-loop policy to predict hand trajectories and desired contact forces, which are then retargeted to a robot using a PD controller that enables precise and robust force control. Through experiments across diverse manipulation tasks, we demonstrate that FTF significantly outperforms prior baselines and and remains robust under perturbations. These results highlight the power of modeling human tactile behavior. Existing limitations of FTF include: 1) FTF aggregates shear and normal forces, losing directional detail; future work could separate and stabilize force components for more dexterous tasks. 2) data collection currently relies on fixed, calibrated cameras. Using egocentric views and stereo triangulation may enable in-the-wild deployment.

References

[1] Raunaq Bhirangi, Venkatesh Pattabiraman, Enes Erciyes, Yifeng Cao, Tess Hellebrekers, and Lerrel Pinto. Anyskin: Plug-and-play skin sensing for robotic touch, 2024. URL https://arxiv.org/abs/2409.08276.

[2] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[3] Siddhant Haldar and Lerrel Pinto. Point policy: Unifying observations and actions with key points for robot manipulation, 2025. URL https://arxiv.org/abs/2502.20391.

[4] Siddhant Haldar, Zhuoran Peng, and Lerrel Pinto. Baku: An efficient transformer for multi-task policy learning, 2024. URL https://arxiv.org/abs/2406.07539.

[5] Aadhithya Iyer, Zhuoran Peng, Yinlong Dai, Irmak Guzey, Siddhant Haldar, Soumith Chintala, and Lerrel Pinto. Open teach: A versatile teleoperation system for robotic manipulation, 2024. URL https://arxiv.org/abs/2403.07870.

[6] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together, 2024. URL https://arxiv.org/abs/2307.07635.

[7] Alexander Khazatsky. Droid: A large-scale in-the-wild robot manipulation dataset, 2024. URL https://arxiv.org/abs/2403.12945.

[8] Mara Levy, Siddhant Haldar, Lerrel Pinto, and Abhinav Shirivastava. P3-po: Prescriptive point priors for visuo-spatial generalization of robot policies, 2024. URL https://arxiv.org/abs/2412.06784.

[9] Fangchen Liu, Chuanyu Li, Yihua Qin, Ankit Shaw, Jing Xu, Pieter Abbeel, and Rui Chen. Vitamin: Learning contact-rich tasks through robot-free visuo-tactile manipulation interface, 2025. URL https://arxiv.org/abs/2504.06156.

[10] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*, 2019.

[11] Venkatesh Pattabiraman, Yifeng Cao, Siddhant Haldar, Lerrel Pinto, and Raunaq Bhirangi. Learning precise, contact-rich manipulation through uncalibrated tactile skins, 2024. URL https://arxiv.org/abs/2410.17246.

[12] Carmelo Sferrazza, Younggyo Seo, Hao Liu, Youngwoon Lee, and Pieter Abbeel. The power of the senses: Generalizable manipulation from vision and touch through masked multimodal learning, 2023. URL https://arxiv.org/abs/2311.00924.

[13] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion, 2023. URL https://arxiv.org/abs/2306.03881.

[14] Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and C. Karen Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation, 2024. URL https://arxiv.org/abs/2403.07788.

[15] William Xie and Nikolaus Correll. Towards forceful robotic foundation models: a literature survey, 2025. URL https://arxiv.org/abs/2504.11827.

[16] Kelin Yu, Yunhai Han, Qixian Wang, Vaibhav Saxena, Danfei Xu, and Ye Zhao. Mimictouch: Leveraging multi-modal human tactile demonstrations for contact-rich manipulation, 2025. URL https://arxiv.org/abs/2310.16917.

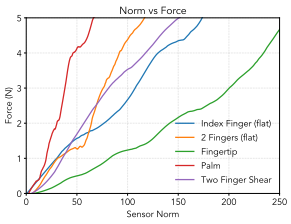
[17] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[18] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. *CoRR*, abs/1812.07035, 2018. URL http://arxiv.org/abs/1812.07035.

V. Appendix

A. FTF

Sensor Norm to Force (Newton) conversion In order to demonstrate a transform between the sensor norm and applied force, we collect data by pressing on an AnySkin sensor mounted on a weighing scale and record synchronous data from both the sensor and the scale streamed through USB at 10Hz. We press the sensor in 5 different manners gradually increasing the force from 0 to 5N, in order to capture different pressures and diverse modes of contact. The sensor norm to applied force comparison is illustrated in Figure 2.



(a) Transformation between the sensor norm and applied force, for various modes of contact, ranging from low-pressure (Palm) to high pressure (Index Fingertip)

(b) Different modes of contact for data collection; Two fingers laid flat (top left), index fingertip (bottom left), palm (middle), one finger laid flat (top right) and two fingers pressing at an angle to emulate a combination of both normal and shear forces

Fig. 2: (Left) Mapping between sensor norm and applied force across various contact modes. (Right) Data collection setup illustrating different types of contact used in the mapping process.

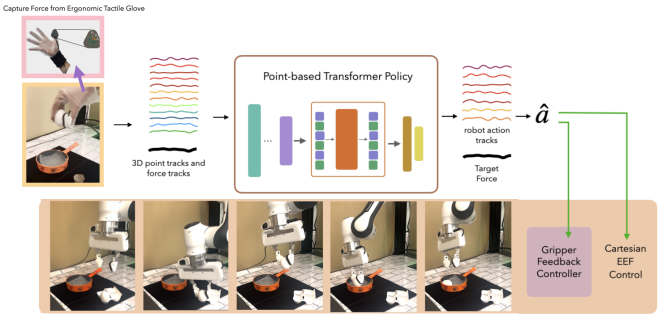


Fig. 3: FTF allows zero-shot transfer of tactile human demonstrations to a Franka Robot.

Embodiment Agnostic Scene Representation The human hand motion data from tactile gloves is converted into a point-based representation to enable robot policy learning from human demonstrations.

1) Human-to-Robot Embodiment Transfer: For each time step t of a human video, we use Mediapipe [10] to extract image key points p_h^t on the human hand. Using point triangulation, the corresponding hand key points from two fixed, calibrated camera views are projected to 3D hand key points $\mathcal{P}_h^t$. We use point triangulation for 3D projection due to its higher accuracy as compared to sensor depth from the camera [3]. The robot position $\mathcal{R}_{pos}^t$ is computed as the midpoint between the tips of the index finger and thumb in $\mathcal{P}_h^t$. The robot orientation $\mathcal{R}_{ori}^t$ is computed as

$$\begin{aligned} \Delta\mathcal{R}_{ori}^t &= \mathcal{T}(\mathcal{P}_h^0, \mathcal{P}_h^t) \\ \mathcal{R}_{ori}^t &= \Delta\mathcal{R}_{ori}^t \cdot \mathcal{R}_{ori}^0 \end{aligned} \tag{1}$$

where $\mathcal{T}$ computes the rigid transform between hand key points on the first frame of the video, $\mathcal{P}_h^0$, and $\mathcal{P}_h^t$. The robot end effector pose is then represented at $T_r^t \leftarrow \{\mathcal{R}_{pos}^t, \mathcal{R}_{ori}^t\}$. Finally, the robot pose T_r^t is converted to N robot key points through a set of N rigid transformations T about the computed robot pose such that

$$(\mathcal{P}_r^t)^i = T_r^t \cdot T^i, \quad \forall i \in \{1, ..., N\} \tag{2}$$

The robot's gripper state $\mathcal{R}_g$ is considered closed when the distance between the tip of the index finger and thumb is less than 7cm, otherwise open. The continuous force value measured for each step, $\mathcal{R}_f^t$, is also included in the robot state. This process has been illustrated in Figure 3.

2) Scene Key Point Representation: The environment is represented as key points through sparse human annotations, following prior work [8, 3]. Given a single demonstration frame, a human user annotates semantically meaningful key points on task-relevant objects in the scene. Using DIFT [13], an off-the-shelf semantic correspondence model, the annotations are propagated to the first frames of all other demonstrations, minimizing human effort. For each demonstration, Co-Tracker [6], an off-the-shelf point tracker, then tracks the initialized key point through each trajectory, efficiently handling occlusions and maintaining temporal consistency. To obtain 3D object key points, we triangulate the tracked key points from the two camera views, grounding them in the robot's base frame. During inference, DIFT is used to localize keypoints in the first frame, after which Co-Tracker tracks them during execution. This approach leverages large pre-trained vision models to generalize across novel object instances and scenes without additional training, requiring only a single frame of user input per task.

Policy Learning For policy learning, we use a transformer policy architecture [4, 3] that takes as input the robot points $\mathcal{P}_r$ and object points $\mathcal{P}_o$ along with the binarized gripper state $\mathcal{R}_g$ and continuous force value $\mathcal{R}_f$. Since the gripper state and force value are 1D and the points are 3D, we repeat the value 3 times when appending to the point tracks to ensure dimensional consistency. A history of observations for each key point is flattened into a single vector and encoded using a multilayer perceptron (MLP) encoder. Each encoded point track and the history of gripper and force values are fed as a separate token into the transformer policy, which predicts the future tracks for each robot point $\hat{\mathcal{P}}_r$, the robot gripper state $\hat{\mathcal{G}}_r$, and future gripper force predictions $\hat{\mathcal{F}}_r$. using a deterministic action head. Following prior works in policy learning [17, 2], we use action chunking with exponential temporal averaging to ensure temporal smoothness of the predicted point tracks. The policy is trained using a mean

squared error loss. The transformer is non-causal in this scenario, and the training loss is only applied to the robot point tracks.

Inference Algorithm 1: FORCEFEEDBACKGRIPPERCONTROL($\hat{F}_t$)

Algorithm 1

Initialize $\tau \leftarrow 0$
repeat
  $\Delta g_t^\tau = k \cdot (\hat{F}_t - F_t^\tau)$
  $g_t^{\tau+1} = g_t^\tau + \Delta g_t^\tau$
  Execute gripper action $g_t^{\tau+1}$
  Read $F_t^{\tau+1}$ from AnySkin
  $\tau \leftarrow \tau + 1$
until $||\hat{F}_t - F_t^\tau|| \leq \epsilon$

a) Robot pose from predicted key points: The predicted robot points $\hat{\mathcal{P}}_r$ are mapped back to the robot pose using constraints from rigid-body geometry. We first consider the key point corresponding to the robot's wrist $\hat{\mathcal{P}}_r^{wrist}$ as the robot position $\hat{\mathcal{R}}_{pos}$. The robot orientation $\hat{\mathcal{R}}_{ori}$ is computed using Eq. 1 considering $\mathcal{R}_{ori}^0$ is fixed and known. Finally, the robot pose $\hat{\mathcal{R}}_{pose}$ is defined as $(\hat{\mathcal{R}}_{pos},\ \hat{\mathcal{R}}_{ori})$.

Algorithm 2 FTF Policy Inference

Obtain object keypoints on first frame using DiFT on annotated dataset frame.
for t in rollout **do**
  Compute action chunk $(\hat{\hat{a}}_t, ..., \hat{\hat{a}}_{t+H}) = \pi(a|s_t)$ and obtain $\hat{a}_t$ with temporal aggregation.
  Parse action: $(\hat{F}_t, \hat{g}_t, \hat{a}_t^{eef}) \leftarrow \hat{a}_t$
  if $\hat{g}_t > closethreshold$ **then**
    Call FORCEFEEDBACKGRIPPERCONTROL($\hat{F}_t$)
  else if $\hat{g}_t < openthreshold$ **then**
    Open gripper
  end if
  Execute $\hat{a}_t^{eef}$ on robot
  Read next state s_{t+1} using Co-Tracker
end for

b) Inference-time PD force controller: To deploy the tactile policy on the robot arm, we need a means for the robot gripper exerting the force predicted by the policy at each step. For this, we design an outer-loop PD controller that adjusts the target gripper closure setpoints to stabilize the measured forces. If at some timestep t, the policy predicts a force $\hat{F}_t$ to be applied, the controller is:

$$\Delta g_t^\tau = k \cdot (\hat{F}_t - F_t^\tau) \tag{3}$$

where τ is the inner loop timestep of the PD controller and F_t^τ is the force read by the robot at timestep t of the policy and timestep τ of the controller. At each step the gripper closure is updated as $g_t^{\tau+1} = g_t^\tau + \Delta g_t^\tau$.

The PD controller runs until the convergence condition $||\hat{F}_t - F_t^\tau|| < \epsilon$. After the controller converges to the desired $\hat{F}_t$, the policy predicts the next action for step $t+1$. We find $k = 0.001$ and $\epsilon = 5$ to work well across all tasks. Finally, the action $\hat{\mathcal{A}}_r = (\hat{\mathcal{R}}_{pose}, \hat{\mathcal{G}}_r, g_t)$ is executed on the robot using end-effector position control at a 6Hz frequency.

B. Experiments

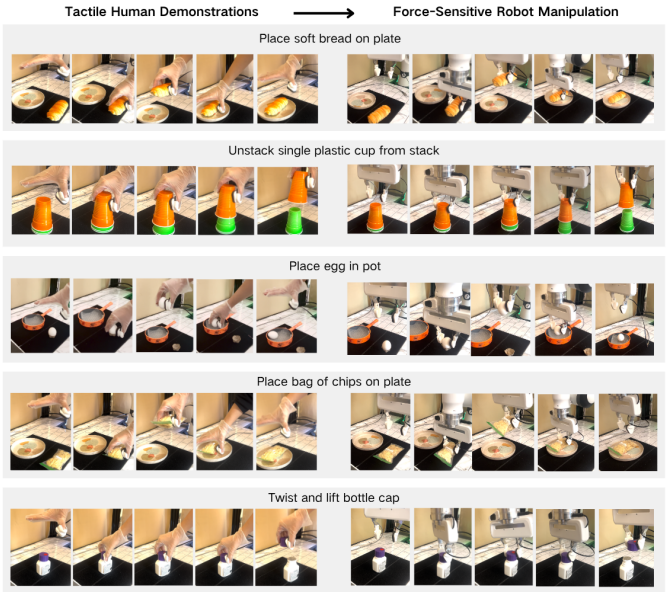
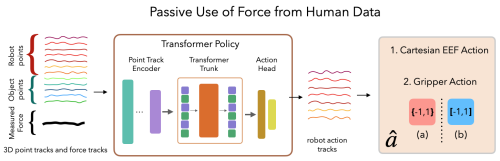


Fig. 4: Visual comparison of tactile human demonstrations (left) and force-sensitive robot manipulation rollouts (right) learned from the human demonstrations.

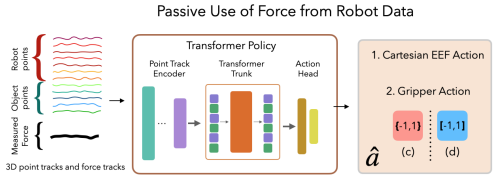
Experimental Setup We evaluate FTF on a Franka Panda robot, operating in a real-world tabletop manipulation environment. Two Intel RealSense D435 cameras are mounted to provide third-person RGB images to our policy. For baselines we also collect 30 demonstrations on the Franka robot per task using a VR-based teleoperation framework [5]. Demonstrations are recorded at 20Hz and subsampled to approximately 6Hz. For methods outputting robot actions, we use absolute actions with orientation represented with a 6D rotation representation [18].

Task descriptions

- `Place soft bread on plate`: Pick and place a deformable bread slice without crushing it.
- `Unstack single plastic cup from stack`: Isolate and lift a single plastic cup from a stack.
- `Place egg in pot`: Delicately place an egg into a pot without breakage.
- `Place bag of chips on plate`: Move a transparent chip bag while preserving its contents.
- `Twist and lift bottle cap`: Twist and remove a cap without disturbing the bottle.



(a) Point track baselines from human data with passive use of force. (a) uses a binary gripper action space by thresholding human hand closure and (b) retargets continuous human hand closure to continuous gripper.



(b) Action imitation baselines from robot data with passive use of force. (c) uses a binary gripper action space and (d) uses continuous gripper action space.

Fig. 5: Comparison of gripper action space across human (left) and robot (right) baselines under passive force conditions.

Baselines We compare FTF with 5 baselines - *Tactile Point Policy* [3], *Continuous-Gripper Tactile Point Policy*, *FTF + Tactile P3-PO* [8], *Tactile P3-PO*, and *Continuous-Gripper Tactile P3-PO*. We describe each method below. We compare FTF with 5 baselines - *Tactile Point Policy* [3], *Continuous-Gripper Tactile Point Policy*, *FTF + Tactile P3-PO* [8], *Tactile P3-PO*, and *Continuous-Gripper Tactile P3-PO*. We describe each method below.

a) *Tactile Point Policy* [3] performs behavior cloning from point tracks extracted from human data as well as force readings from the tactile glove and predicts future tracks which are converted into robot actions. This baseline provides a comparison to methods such as [11] that use force input to improve the precision of learned policies but in the context of human data.

(b) *Continuous-Gripper Tactile Point Policy* is similar to *Tactile Point Policy* but predicts continuous gripper closure. The gripper closure value is measured as the distance between the index and thumb tracked points from the human data re-normalized to the range of the robot gripper.

(c) *FTF + Tactile P3-PO* extends *Tactile P3-PO* by predicting both robot actions and future contact forces. The model is trained on teleoperated robot data, using force signals collected during teleoperation as input, and outputs predicted forces alongside actions. This baseline evaluates whether incorporating force prediction improves control performance in robot teleoperation setting and compares the utility of robot-collected versus human-collected force data.

(d) *Tactile P3-PO* [8] predicts teleoperated robot actions from robot tracks obtained by unprojecting robot and object points of interst into 3D space. The method also inputs force readings from the robot gripper collected during teleoperation into the Transformer policy. This method provides a similar comparison to *Tactile Point Policy* but on teleoperated robot data and ground truth robot actions.

(e) *Continuous-Gripper Tactile P3-PO* is similar to *Tactile P3-PO* but predicts continuous gripper closure. The continuous gripper values are obtained directly from the robot gripper during teleoperator using an adaptation to the VR-based teleoperation framework [5] that allows the teleoperator to output continuous gripper closures based on visual feedback during data collection.

TABLE III: FTF performance under test-time disturbance

Task	FTF
Place bag of chips on plate	10/15

FTF is robust to test-time disturbances. In the *placing a bag of chips on a plate* task, we introduce disturbances such as holding the bag down or pressing during lift. Despite changes in force profiles, FTF adapts and maintains a 67% success rate (Table III).

TABLE IV: Performance comparison of masked vs umasked force tracks inputted to FTF

Task	Masked Force	Unmasked Force
`Lift and hold bread`	10/10	10/10

FTF does not require force input to perform effectively. We implemented a variant of FTF that masks the force data fed to the Transformer, constraining the model to predict desired forces solely based on the environment and robot state. We evaluate FTF on a simple task `Lift and hold bread`, involving picking up a soft piece of bread without crushing it and suspending the grasp in the air in Table IV. This modification did not result in any degradation of force prediction or task performance, suggesting that FTF can achieve effective force control even without explicit force input.