Appendix

AI. Proof of Propositions

In this section, we present the detailed proofs omitted from the main manuscript. We begin by introducing some preliminaries.

A. Preliminaries

We are interested in the matrix Lie group representation g of the manipulator's end-effector pose, where $g \in SE(3)$, given by

$$g = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \in SE(3), \tag{12}$$

where $SE(3)$ is a Special Euclidean group, $R \in SO(3)$ with $SO(3)$ being a Special Orthogonal group, and $p \in \mathbb{R}^3$. We first define invariance and equivariance.

Definition 1 ($SE(3)$ left invariance and equivariance)**.** Let f be a function $f : \mathcal{X} \to \mathcal{Y}$, so that $y = f(x)$, where $x \in \mathcal{X}$ is a domain and $y \in \mathcal{Y}$ is a co-domain. Then, a function f is left-invariant to $SE(3)$ (left) group action $g_l \in SE(3)$ if the following equation holds:

$$f(g_l \circ x) = f(x), \tag{13}$$

where $\circ$ is a group action on the domain or co-domain .

Similarly, a function f is left-equivariant to $SE(3)$ group action g_l if the following holds:

$$f(g_l \circ x) = g_l \circ f(x). \tag{14}$$

In fact, the group action is realized by the appropriate group representation on the acting set, i.e., the domain or co-domain, which is often denoted by $\rho(g_l)$ in group equivariant deep learning literature [22]. Important examples widely used throughout the paper include cases where the domain or co-domain is $SE(3)$ itself or the wrench[1] $\mathbb{R}^6$. If the set that the group acts on is the $SE(3)$ group itself, then,

$$g_l \circ g = g_l \cdot g, \quad \forall g, g_l \in SE(3), \tag{15}$$

where $\cdot$ is a standard matrix multiplication. If the set that the group actions on is the wrench $\mathbb{R}^6$, then,

$$g_l \circ h = \mathrm{Ad}^T_{g_l^{-1}} h, \quad \forall g_l \in SE(3),\ \forall h \in \mathbb{R}^6, \tag{16}$$

where $\mathrm{Ad} : SE(3) \times \mathbb{R}^6 \to \mathbb{R}^6$, defined as

$$\mathrm{Ad}_{g_l} = \begin{bmatrix} R_l & \hat{p}_l R_l \\ 0 & R_l \end{bmatrix} \in \mathbb{R}^{6\times 6}, \tag{17}$$

with $g_l = (p_l, R_l)$. For the details of the group action and representation, we refer to [22].

Lemma 1 (Left invariance of GCEV and elastic wrench [19])**.** The GCEV $e_{_G}$ (2) and elastic wrench $f_{_G}$ (9) is left-invariant, i.e., $\forall g_1, g_2, g_l \in SE(3)$,

$$\begin{aligned} e_{_G}(g_l \circ (g_1, g_2)) &= e_{_G}(g_l g_1, g_l g_2) = e_{_G}(g_1, g_2), \\ f_{_G}(g_l \circ (g_1, g_2, K_p, K_R)) &= f_{_G}(g_l g_1, g_l g_2, K_p, K_R) \\ &= f_{_G}(g_1, g_2, K_p, K_R). \end{aligned} \tag{18}$$

Proof: Although the full proof is presented in [19], we include it for completeness. Let $g_l = (p_l, R_l)$, and $g_i = (p_i, R_i)$ with $i = \{1, 2\}$. Then, the left-transformed homogeneous matrix $g_l g_i$ is calculated in the following way:

$$g_l g_i = \begin{bmatrix} R_l & p_l \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_i & p_i \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_l R_i & R_l p_i + p_l \\ 0 & 1 \end{bmatrix},$$

i.e., $g_l g_i = (R_l p_i + p_l, R_l R_i)$ The left-transformed GCEV is then

$$\begin{aligned} e_{_G}(g_l g_1, g_l g_2) &= \begin{bmatrix} R_1^T R_l^T (R_l p_1 + p_l - R_l p_2 - p_l) \\ ((R_l R_2)^T R_l R_1 - (R_l R_1)^T R_l R_2)^\vee \end{bmatrix} \\ &= \begin{bmatrix} R_1^T (p_1 - p_2) \\ (R_2^T R_1 - R_1^T R_2)^\vee \end{bmatrix} = e_{_G}(g_1, g_2), \end{aligned} \tag{19}$$

where the definition of the rotation matrix is used, i.e., $R^T R = RR^T = I, \forall R \in SO(3)$. Similarly, the left-transformed elastic wrench reads

$$\begin{aligned} & f_{_G}(g_l g_1, g_l g_2, K_p, K_R) \\ &= \begin{bmatrix} (R_l R_1)^T R_l R_2 K_p (R_l R_2)^T (R_l p_1 + p_l - R_l p_2 - p_l) \\ (K_R (R_l R_2)^T R_l R_1 - (R_l R_1)^T R_l R_2 K_R)^\vee \end{bmatrix} \\ &= \begin{bmatrix} R_1^T R_2 K_p R_2 (p_1 - p_2) \\ (K_R R_2^T R_1 - R_1^T R_2 K_R)^\vee \end{bmatrix} = f_{_G}(g_1, g_2, K_p, K_R). \end{aligned} \tag{20}$$

■

We note that the gains K_p and K_R are defined on the desired frame [2], i.e., on the body-frame; therefore, they are not affected by left-group actions (change of spatial coordinate system).

B. Proof of Proposition 1

The left-transformed observation signals $g_l \circ o(k)$ reads that:

$$g_l \circ o(k) = (g_l \circ e_{_G}, g_l \circ F_e, g_l \circ I_w). \tag{21}$$

As was shown in Lemma 1, the GCEV $e_{_G}$ is left invariant as

$$g_l \circ e_{_G}(g, g_{_{EDF}}) = e_{_G}(g_l g, g_l g_{_{EDF}}) = e_{_G}(g, g_{_{EDF}}).$$

The force-torque sensor values are left-invariant because they are already defined with respect to the end-effector frame [19], and the visual representation vectors satisfy left invariance if Assumption 1 is ideally met. Combining all these properties, it follows that

$$\begin{aligned} a(k) &= \pi_\theta(g_l \circ o(k)) = \mathcal{D}_\psi(g_l \circ e_{_G}, g_l \circ F_e, \mu_\phi(g_l \circ I_w)) \\ &= \mathcal{D}_\psi(e_{_G}, F_e, z) = \pi_\theta(o(k)), \end{aligned} \tag{22}$$

which shows the left invariance of the G-CompACT policy on the end-effector frame. ■

C. Proof of Corollary 1

Notice that the desired pose signal in the spatial frame g_d is obtained via (with a slight abuse of notation)

$$g_d = g g_{rel} = g \cdot \pi_\theta(o(k)) \triangleq \pi^s_\theta(o(k)), \tag{23}$$

[1] Although the original wrench should be represented in $se^*(3)$, a dual-space of Lie-algebra, we use a vector representation of $se^*(3)$ to reduce mathematical details.

where the superscript s denotes that the policy is described on the spatial frame, i.e., the world frame. Then, utilizing the left-invariance property from Proposition 1, it follows that

$$\begin{aligned}(g_l g_d, K_p, K_R) &= \pi^s_\theta(g_l \circ o(k)) \\ &= g_l g \cdot \pi(g_l \circ o(k)) = g_l g \cdot \pi(o(k)).\end{aligned} \tag{24}$$

Therefore, when the policy is left-transformed by an arbitrary element $g_l \in SE(3)$, the resulting trajectories in the spatial frame g_d are also transformed to $g_l g_d$, showing the equivariance property. ■

D. Proof of Proposition 2

Let the object of interest, e.g., a peg for the picking task and a hole for the placing task, be observed by $\mathcal{O}^{ref}$ and I_w with its pose given by g_{ref}, so that the left-translated $g_l \cdot g_{ref}$ is observed by $g_l \circ \mathcal{O}^{ref}$ from the point cloud, and $g_l \circ I_w$ by the left-translated end-effector attached wrist camera as described in Fig. 5. First, notice that h_Θ can be fully written as

$$\begin{aligned}h_\Theta(g, g_{ref}, F_e) &= f_{_G}(g, g_d, K_p, K_R) \\ &= f_{_G}(g, \pi^s_\theta(e_{_G}(g, g_{_{EDF}}), F_e, I_w)) \\ &= f_{_G}(g, \pi^s_\theta(e_{_G}(g, f_\varphi(\mathcal{O}^{ref})), F_e, I_w))\end{aligned} \tag{25}$$

Note also that $g_{_{EDF}}$ is fed to $e_{_G}$, not g_{ref}.

Then, when both g and g_{ref} undergo a left transformation g_l, from Assumption 1 and Corollary 1, the following holds:

$$\begin{aligned}&h_\Theta(g_l g, g_l g_{ref}, g_l \circ F_e) \\ &= f_{_G}(g_l g, \pi^s_\theta(e_{_G}(g_l g, f_\varphi(g_l \circ \mathcal{O}^{ref})), g_l \circ F_e, g_l \circ I_w)) \\ &= f_{_G}(g_l g, \pi^s_\theta(e_{_G}(g_l g, g_l g_{_{EDF}}), g_l \circ F_e, g_l \circ I_w)) \\ &= f_{_G}(g_l g, g_l g_d, K_p, K_R) = f_{_G}(g, g_d, K_p, K_R) \\ &= h_\Theta(g, g_{ref}, F_e).\end{aligned} \tag{26}$$

The second-last equation ($SE(3)$ left-invariance of the elastic wrench) comes from Lemma 1. Finally, as the h_Θ is left-invariant and is defined on the end-effector frame, from the result of Proposition 2 of [19], h_Θ is equivariant, if it is described in the spatial frame, i.e.,

$$h^s_\Theta(g_l g, g_l g_{ref}, g_l \circ F_e) = \mathrm{Ad}^T_{g_l^{-1}} h_\Theta(g, g_{ref}, F_e), \tag{27}$$

where Ad is a (large) adjoint operator. From Definition 1, h^s_Θ is an equivariant function [19]. ■

AII. Implementation Details

A. G-CompACT Training

1) Details on models: The objective of this chapter is to highlight the details of the selected models, especially the vision encoders. Our G-CompACT model has a CLIP-RN50 vision backbone that is modulated by the FiLM layer from the CLIP text encoder. A few notable hyperparameters are summarized in the Table A1. As denoted in Fig. 1, we use 30Hz of inference frequency, with the chunking size of 60. In addition, we used rotation vector (`rotvec`) representation for relative pose actions, and 6D rotation (`rot6d`) representation for world-pose observation and actions for benchmark models. This is because the default end-effector configuration in the world (spatial) frame tends to have a 180° rotation angle, leading to a sign flip when using the rotation vector representation.

TABLE A1: Hyperparameters of G-CompACT. The other hyperparameters are adapted from the ACT [37].

Names	Values
Image Size	$[224, 224]$
Learning Rate (policy) η_{policy}	$1e-05$
Learning Rate (Vision Encoder) η_{vision}	$1e-05$
Epochs	$15,000$
Batch Size	32
Batch Size	32

2) Dataset details: To collect the demonstration dataset, the expert teleoperator monitors the task's progress, makes real-time movement commands via a SpaceMouse, and adjusts the admittance gains using keyboard input to switch between pre-defined gain modes: low-gain mode, high-gain mode, insertion mode, and contact mode.

a) Admittance Gains: The gain modes utilized during the training are low-gain mode, high-gain mode, insertion mode, and contact mode. The low/high gain mode has low/high gains in all directions, the insertion mode has high gains in the z direction of the end-effector frame and low gains elsewhere. Finally, the contact mode has low gains in the z direction and high gains elsewhere. In our EquiContact implementation, we use $M = 0.5 I_{6\times 6}$. We used only the diagonal terms of the stiffness matrices K_p and K_R for learning and GAC implementation[2]. Therefore, the stiffness gains (K_p, K_R) can be represented with 6 dimension. The details are as follows:

- Low-gain: $(K_p, K_R) = (300, 300, 300, 300, 300, 300)$
- High-gain: $(K_p, K_R) = (1000, 1000, 1000, 1000, 1000, 1000)$
- Contact mode: $(K_p, K_R) = (1500, 1500, 300, 1500, 1500, 1500)$
- Insertion mode: $(K_p, K_R) = (300, 300, 1500, 300, 300, 300)$

We note that the detailed gains implementation may vary significantly depending on the specific implementation, such as sampling frequency and even robot firmware version. The sets of working gains are found through trial and error.

b) Data Collection Methods: For the PiH task, the end-effector is first aligned with high-gain mode, quickly converted to the contact mode to make a surface contact and search for a hole, and the insertion mode is activated when the peg is slightly inserted into the hole. For the surface wiping task, the high-gain mode is used to align with the whiteboard, and the surface contact mode is used during the surface wiping. For the screwing task, the high-gain mode is first used to align with the screw hole, followed by contact mode for fine searching. Then, the insertion mode is activated to ensure screw-locking, and the low-gain mode is used for screw rotation.

c) Scene Randomization and details: The initial pose of the end-effector is randomized for both with and without background variations. We add arbitrary lab objects with random poses for the background variations. The examples of the scene randomization for PiH are presented in Fig. A1.

3) Text Prompts: Here, we additionally provide the full text prompts used during the training. As mentioned in Sec. IV,

[2]This is also a great benefit of using geometric impedance/admittance control [20].

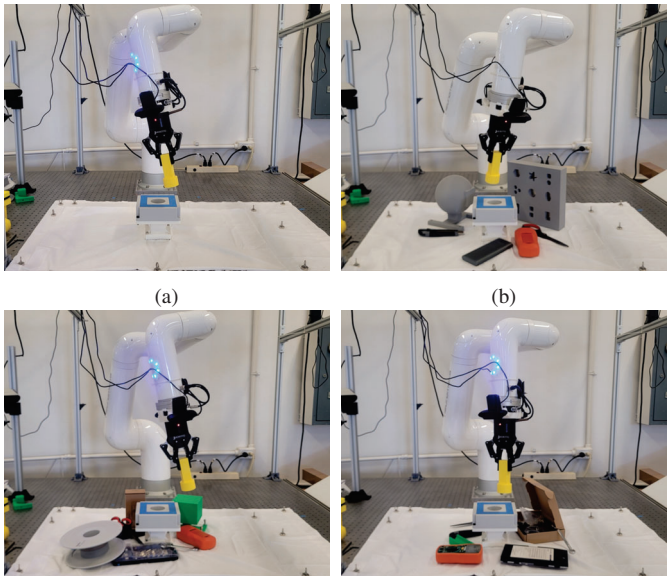

Fig. A1: Examples of scene randomization during data collection are shown. Notice the initial pose randomization of the end-effector for all cases. (a) Example without background variations. (b)-(d) Examples with background variations composed of arbitrary lab objects with random poses.

the average text tokens are fed during the inference phase. We provide the full text prompts for the PiH-placing task, but show few samples for the other tasks to reduce verbosity.

a) PiH - Placing: Below is the full sets of text prompts utilized for the PiH placing task.

- "A yellow peg approaching a light-gray assembly target marked with blue."
- "A cylindrical yellow peg inserted into a round light-gray hole on a square board with blue tape edges."
- "A yellow dowel being aligned with a light-gray target that has blue markings on a flat surface."
- "A yellow peg going into a light-gray round hole with blue tape border."
- "A yellow cylindrical peg and a round light-gray hole with blue tape around the square board."
- "A yellow plastic dowel aligning with a round light-gray socket, blue tape square."
- "A thick yellow pin approaching a light-gray round socket with blue tape."
- "A yellow peg being inserted into a light-gray circular hole on a board with blue tape."
- "A thick yellow stick above a light-gray round socket with blue tape."
- "A thick light-yellow stick aligning to a light-gray assembly target with blue marks."
- "A yellow plastic dowel being inserted into a round socket with blue tape on a flat surface."
- "Peg-in-hole task: yellow plastic peg and light-gray round hole with blue tape around the board."
- "A cylindrical yellow peg entering a light-gray circular recess with a blue tape border."
- "A yellow peg being aligned with a light-gray target that has blue markings on a flat surface."
- "A yellow peg approaching a light-gray assembly target marked with blue."
- "A yellow cylindrical peg inserted into a round light-gray hole on a square board with blue tape edges."
- "A yellow peg reoriented to align with a light-gray circular hole bordered with blue tape."
- "A yellow cylindrical peg re-aligning to fit into a round light-gray hole with blue tape edges."

b) PiH - Picking: Among the 13 prompts, we present 3 prompts for example in this paper.

- "A black robotic gripper is about to pick up a yellow peg."
- "A robotic pick-up: a black gripper and a yellow peg object."
- ⋮
- "A black robotic gripper reaching toward a yellow cylindrical dowel"

c) Screwing Task: Below is the task prompts example of screwing task among 16 prompts.

- "A yellow cylindrical peg rotating inside a round light-gray hole with blue tape border."
- "yellow plastic dowel aligning with a round light-gray socket, blue tape square."
- ⋮
- "A yellow peg being aligned and screwed into a light-gray target with blue markings."

d) Surface Wiping Task: Below is the task prompts example of surface wiping task among 16 prompts.

- "A black metallic robotic gripper wiping black markings with a yellow eraser."
- "robotic gripper grasping a yellow eraser moving on top of the black markings."
- "a black robot gripper holding a yellow eraser moving over black lines."
- ⋮
- "black markings being erased by a yellow eraser held by a robot gripper."

B. Diff-EDF Implementation Details

Instead of following the original Diff-EDF pipeline, which utilizes pick-and-place models, we used two pick models. This decision mainly stems from the task setup, where the peg is upright, and the peg is grasped by the gripper in an aligned, upright pose. The core difference of the pick and place model is that the place model needs to get the grasp point cloud after each grasp to handle the right equivariance of the model. However, from task setup, we bypass this right equivariance issue, removing the necessity of the place model.

We also used the post-processing heuristics to filter the output pose of the Diff-EDF. The Diff-EDF outputs 20 candidate target poses for picking and 20 candidates for the place, which are ranked by the energy level. Although in theory, the lower energy poses should result in a better pose, we found out that this does not hold in practice. Instead, we figured out that the

Algorithm 1 Inference Procedure of EquiContact

Require: Diff-EDF f_{θ_1}, G-CompACT π_{θ_2}, Task $\in$ {pick, place}
Get scene and grasp point cloud $\mathcal{O}^{scene}$, $\mathcal{O}^{grasp}$
Run Diff-EDF for reference frame $g_{EDF} = f_\varphi(\mathcal{O}^{scene}, \mathcal{O}^{grasp})$
Move the end-effector near the reference frame and initialize EquiContact π_{θ_2}
for each inference timestep k **do**
    Get current sensor values $g(k)$, $F_e(k)$, $I_w(k)$
    Calculate GCEV $e_{_G}(k) = e_{_G}(g(k), g_{EDF})$ (2)
    Run G-CompACT:
        $(g_{rel}, K_p, K_R)(k) = \pi_\theta(e_{_G}, F_e, I_w)(k)$ (1)
    Calculate desired pose $g_d(k) = g(k) g_{rel}(k)$
    Update $(g_d, K_p, K_R)(k)$ for GAC loop
    Run GAC realizing desired dynamics (8), (9)
end for

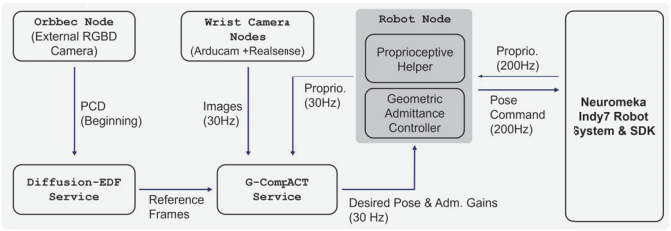

Fig. A2: Whole pipeline implemented with ROS2 is presented.

Diff-EDF have low variations on the position of the "tip". The mean value of the tip position candidates are first calculated and is used to recalculate the pose of the end-effector from the known orientation (upright peg assumption). On the other hand, for the placing, the desired orientation is calculated by taking a mean value of the orientation. The position of the end-effector is similarly calculated from the position of the tip.

C. GAC Implementation

We implement the geometric admittance controller (GAC) using the pose tracking controller. Given the desired dynamics (8), the desired end-effector pose command $\tilde{g}_d(k)$ provided to the end-effector controller is calculated in discrete time as

$$V_d^b(k) = V^b(k) + T_s \cdot M^{-1}(F_e(k) - f_{_G}(k) - K_d V^b(k)),$$
$$\tilde{g}_d(k) = g(k) \cdot \exp(\hat{V}_d^b(k) \cdot T_s), \tag{28}$$

where T_s is a sampling time ($5\mathrm{ms}$ for GAC) and $\hat{(\cdot)}$ denotes a hat-map.

D. Full Pipeline Implementation

As mentioned earlier, the full EquiContact pipeline is implemented with ROS2 framework. The whole implementation flow is presented in Fig. A2, and also summarized as in Algorithm 1.

AIII. Additional Experimental Results

A. Errors of Diff-EDF

The RMSE error of the Diff-EDF on the training dataset is presented in Table A2. The RMSE of the rotational error is naively calculated from the Euler angles of the error rotation matrix.

TABLE A2: RMSE error values of Diff-EDFs on the *training dataset*. The dimensions of translational errors in x, y, z directions, given by $e_{T,x}, e_{T,y}, e_{T,z}$, are mm and the rotational errors in x, y, z directions, given by $e_{R,x}, e_{R,y}, e_{R,z}$, are deg.

	$e_{T,x}$	$e_{T,y}$	$e_{T,z}$	$e_{R,x}$	$e_{R,y}$	$e_{R,z}$
pick	7.173	6.933	6.199	7.650	15.90	15.67
place	13.75	8.241	5.999	3.806	5.560	5.660

TABLE A3: Results of vision encoder design study. OOD case here is a 45° transformation in the y axis.

Backbone	Learning Rate (η_{policy})	Learning Rate (η_{vision})	Success Rate In-dist	Success Rate OOD
RN18	$1e-05$	$1e-05$	10 / 10	6 / 10
CLIP-RN50-frozen	$1e-05$	0	3 / 10	3 / 10
CLIP-RN50-SB	$1e-05$	$1e-06$	10 / 10	0 / 10
CLIP-RN50 (**proposed**)	$1e-05$	$1e-05$	10 / 10	10 / 10

As noticed from the table, the translational error is significantly larger than the desired accuracy of precision of the PiH task $\sim 1\mathrm{mm}$. In addition, the rotational error of the picking task is significantly higher than that of the placing task. Therefore, we use the "upright peg" assumption for the full pipeline implementation.

B. Vision Encoder Design Study

Here, we conduct a controlled comparison of vision encoder variants. To verify the design choices to meet the conditions of Assumption 1, we have trained 4 models with the same training dataset for PiH tasks, which are listed below:

- Baseline ACT architecture that uses ResNet 18 and without language feature (RN18)
- ACT with pretrained CLIP-RN50 but is frozen (CLIP-RN50-frozen)
- ACT with CLIP-RN50, but 10% of learning rate for vision backbone (CLIP-RN50-SB, SB stands for slow backbone training)
- ACT with CLIP-RN50, same learning rate for policy and vision backbone (proposed)

We have tested our models in the in-distribution condition and with a 45° transformation in the y axis, i.e., the third case for extreme task transformations (Fig. 3). The results are summarized in Table A3.

We first observe that the vision encoder without language guidance degrades under the OOD rotation (6/10), although background randomization during data collection partially mitigates background overfitting. In contrast, using a frozen CLIP-RN50 encoder yields low success even in-distribution (3/10), suggesting a significant domain mismatch between internet-scale pretraining and the short-range wrist-camera viewpoint in contact-rich manipulation. Interestingly, fine-tuning the CLIP-RN50 encoder with a very small learning rate achieves high in-distribution performance (10/10) but fails completely under the OOD rotation (0/10). We speculate that the visual representation is not sufficiently adapted: the encoder adjusts only locally to the training task configuration, without acquiring robustness

to large geometric shifts, making the downstream policy brittle when viewpoint changes substantially. Finally, jointly fine-tuning the CLIP-RN50 encoder together with the policy (proposed) recovers both in-distribution and OOD performance (10/10), indicating that stronger encoder adaptation is critical for wrist-camera generalization under task transformations.