Appendix

A Motion-invariant Loss

The motion-invariant transform $\phi(\cdot)$, used to compute $\mathcal{L}_{\text{invar}}$ in Equation 5, follows the DHB motion-invariant framework [2]. Given trajectories $\{u_k\}_{k=t-T}^{t}$ with $T \geq 2$, we compute the relative position p_k and orientation r_k of the gripper with respect to the initial frame at $t-T$, where p_{t-T} and r_{t-T} are at the origin.

The differences $\Delta\mathbf{p}_k = \mathbf{p}_{k+1} - \mathbf{p}_k$ and $\Delta\mathbf{r}_k = \mathbf{r}_{k+1} - \mathbf{r}_k$ represent the linear and angular trajectory changes between $k+1$ and k. The initial linear frames are defined as:

$$\begin{aligned}
\hat{\mathbf{x}}_{p,k} &= \frac{\Delta\mathbf{p}_k}{\|\Delta\mathbf{p}_k\|},\\
\hat{\mathbf{y}}_{p,k} &= \frac{\hat{\mathbf{x}}_{p,k} \times \hat{\mathbf{x}}_{p,k+1}}{\|\hat{\mathbf{x}}_{p,k} \times \hat{\mathbf{x}}_{p,k+1}\|},\\
\hat{\mathbf{z}}_{p,k} &= \hat{\mathbf{x}}_{p,k} \times \hat{\mathbf{y}}_{p,k}.
\end{aligned}$$

Similarly, the initial angular frames are:

$$\begin{aligned}
\hat{\mathbf{x}}_{r,k} &= \frac{\Delta\mathbf{r}_k}{\|\Delta\mathbf{r}_k\|},\\
\hat{\mathbf{y}}_{r,k} &= \frac{\hat{\mathbf{x}}_{r,k} \times \hat{\mathbf{x}}_{r,k+1}}{\|\hat{\mathbf{x}}_{r,k} \times \hat{\mathbf{x}}_{r,k+1}\|},\\
\hat{\mathbf{z}}_{r,k} &= \hat{\mathbf{x}}_{r,k} \times \hat{\mathbf{y}}_{r,k}.
\end{aligned}$$

The directions of the axes in both frames are chosen to prevent discontinuities across time steps.

In the DHB transformation, the motion of a rigid body is separated into position and orientation using two frames. The two invariants are the norms of the relative positions and orientations between frames:

$$\begin{aligned}
m_{p,k} &= \|\Delta\mathbf{p}_k\|,\\
m_{r,k} &= \|\Delta\mathbf{r}_k\|.
\end{aligned}$$

These invariants, m_p and m_r, describe the translation of the linear and angular frames. Four additional values describe their rotation:

$$\begin{aligned}
\theta^1_{p,k} &= \arctan\left(\frac{\hat{\mathbf{x}}_{p,k} \times \hat{\mathbf{x}}_{p,k+1}}{\hat{\mathbf{x}}_{p,k} \cdot \hat{\mathbf{x}}_{p,k+1}} \cdot \hat{\mathbf{y}}_{p,k}\right),\\
\theta^2_{p,k} &= \arctan\left(\frac{\hat{\mathbf{y}}_{p,k} \times \hat{\mathbf{y}}_{p,k+1}}{\hat{\mathbf{y}}_{p,k} \cdot \hat{\mathbf{y}}_{p,k+1}} \cdot \hat{\mathbf{x}}_{p,k+1}\right),\\
\theta^1_{r,k} &= \arctan\left(\frac{\hat{\mathbf{x}}_{r,k} \times \hat{\mathbf{x}}_{r,k+1}}{\hat{\mathbf{x}}_{r,k} \cdot \hat{\mathbf{x}}_{r,k+1}} \cdot \hat{\mathbf{y}}_{r,k}\right),\\
\theta^2_{r,k} &= \arctan\left(\frac{\hat{\mathbf{y}}_{r,k} \times \hat{\mathbf{y}}_{r,k+1}}{\hat{\mathbf{y}}_{r,k} \cdot \hat{\mathbf{y}}_{r,k+1}} \cdot \hat{\mathbf{x}}_{r,k+1}\right).
\end{aligned}$$

This process produces the linear and angular invariant values $(m_{p,k}, \theta^1_{p,k}, \theta^2_{p,k})$ and $(m_{r,k}, \theta^1_{r,k}, \theta^2_{r,k})$, as established in the original work.

To ensure continuity, the computed frame rotations are transformed with $\sin(\cdot)$ and $\sin(2\cdot)$. The final transformation used in our regularization term is thus:

$$\phi\left(\{u_k\}_{k=t-T}^{t}\right) = \left\{ \begin{bmatrix} m_{p,k} \\ \sin(\theta_{p,k}^1) \\ \sin(2\theta_{p,k}^1) \\ \sin(\theta_{p,k}^2) \\ \sin(2\theta_{p,k}^2) \\ m_{r,k} \\ \sin(\theta_{r,k}^1) \\ \sin(2\theta_{r,k}^1) \\ \sin(\theta_{r,k}^2) \\ \sin(2\theta_{r,k}^2) \end{bmatrix} \right\}_{k=t-T}^{t-2},$$

yielding 10 variables with a length of $T-1$. When computing $\mathcal{L}_{\text{invar}}$, we use transformed values from two types of trajectories: 1) $\phi(\{\hat{u}_k\}_{k=t-T}^{t})$, the transformed values from the demonstration trajectories, and 2) $\phi(u_t, \{\hat{u}_k\}_{k=t-T}^{t-1})$, the transformed values from the given previous trajectories $\{\hat{u}_k\}_{k=t-T}^{t-1}$ and the predicted target u_t at time t. By calculating the L2 loss between these two transformed values and using it as a training loss, the predicted trajectories u_t are aligned with the demonstration trajectories in the motion-invariant space, given $\{\hat{u}_k\}_{k=t-T}^{t-1}$.

B Implementation Details

The visuomotor policy π_H predicts target poses for the handheld gripper at 10 Hz. The IK optimization π_L realizes these target poses by retargeting them into whole-body motions, updating target joint positions and body orientation at 100 Hz. In simulation, we applied low-level PD control for each joint and body at 500 Hz. For *Spot*, we additionally computed joint positions for the legs by solving IK analytically based on the target body pose. For *Google*, body motion was controlled similarly to other arm joints with PD control, though using high gains. In real robot setups, we controlled the robots through APIs provided by the manufacturers. For quantitative evaluation on *Panda*, we used `JOINT_IMPEDANCE` mode via Deoxys [3] for joint position control. In the demonstration on *Spot*, we directly streamed one-point trajectories for arm joint positions and body pose through Boston Dynamics' Spot SDK.

C Demonstrations in Simulation

Task demonstrations in simulation use the same tracking camera setup as in real-robot evaluations—a Realsense T265 [1]. To replicate real-world human demonstration behaviors, visual odometry data from the tracking camera is mapped to simulated handheld gripper motions in the *Abstract* embodiment or to IK commands for teleoperated simulation robots. The button interface for triggering grasp actions and recording data is kept consistent with the real-world setup. However, unlike real-world demonstrations, simulation does not require physical interaction with the handheld gripper. Therefore, shared gripper components were removed, and a simplified handle was used to reduce the workload on the human demonstrators.

References

[1] *Intel RealSense SDK*, `https://github.com/IntelRealSense/librealsense`.

[2] D. Lee, R. Soloperto, and M. Saveriano, "Bidirectional invariant representation of rigid body motions and its application to gesture recognition and reproduction," *Autonomous Robots*, 2018.

[3] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu, "Viola: Imitation learning for vision-based manipulation with object proposal priors," in *Conference on Robot Learning*, 2022.