WARP: Whole-Body Retargeting for Learning from Offline Human Demonstrations

Author Names Omitted for Anonymous Review. Paper-ID 17

***Abstract*—Direct transfer from human demonstration to learnable robot action is a crucial step towards scalable whole-body mobile manipulation. While human data scales better than mobile teleoperation, it requires overcoming significant embodiment gaps. Existing retargeting methods yield imprecise or inconsistent solutions, causing action multi-modality that prevents supervised policies from reliably converging. We present Whole-body-Aware Retargeting from human Pose (WARP), an offline pipeline that explicitly models embodiment differences to extract precise, unique whole-body actions. WARP leverages a closed-form Shoulder-Elbow-Wrist (SEW) geometric solver for exact end-effector tracking while preserving whole-body structural intent. Paired with lazy mobile-base control, it extracts accurate, consistent robot trajectories. Evaluations show WARP provides highly reliable data for open-loop real-world replay. To our knowledge, WARP is the first framework to train a whole-body mobile manipulation policy directly from offline human demonstrations—without teleoperation action data—and transfer it zero-shot to the real robot. More details on https://warp-retarget.github.io/.**

***Index Terms*—retargeting, whole-body mobile manipulation, imitation learning**

I. Introduction

Learning whole-body mobile manipulation policies directly from offline human demonstrations is an attractive path toward scale: compared to teleoperation data, human data is cheap to record, needs no robot hardware in the loop, and captures unconstrained whole-body behavior in natural environments. The central difficulty is that a human demonstration is not, by itself, a robot demonstration. Human motion carries task-relevant structure that extends well beyond the hands: how the torso is oriented, how the elbows route around obstacles, when the body shifts to extend reach. However, a retargeter that merely places the robot end-effector near the human hand discards the exact whole-body information that whole-body manipulation depends on. To make human data usable, retargeting must convert human motion into robot actions that are both *physically replayable* on the robot and *statistically learnable* by a policy.

These requirements are far stricter in the offline setting than in online teleoperation. During teleoperation, a human operator closes the loop in real time: if the robot wrist drifts, misses contact, or selects an undesirable inverse-kinematics branch, the operator can compensate by changing their motion [11, 5, 27, 1]. Offline policy learning has no such corrective mechanism. The retargeted trajectory *is* the supervision. This exposes a fundamental tension in whole-body retargeting: the robot must satisfy hard task constraints, especially end-effector poses, while also preserving the demonstrated human body configuration, including torso orientation, shoulder placement, elbow routing, and base motion. Prior offline methods have largely overlooked this tension [22, 19, 12]. This creates two failure modes. First, the solution may be *imprecise*: improving whole-body similarity can degrade end-effector tracking, while prioritizing the end effector can discard the torso, elbow, and base structure that made the manipulation feasible. Either case turns retargeting error into supervision error. Second, the solution may be *inconsistent*: because humanoids are highly redundant, nearly identical human poses can map to different configurations depending on solver initialization or local tradeoffs. Similar observations would be paired with divergent actions, which impedes learning.

We address these issues with **WARP** (Fig. 1), an offline human-to-robot retargeting and policy-learning framework for whole-body mobile manipulation. The key mechanism is the Shoulder-Elbow-Wrist (SEW) representation [16]. Instead of directly optimizing over redundant joint angles, SEW represents the arm through its shoulder, elbow, wrist, and hand-frame geometry (Fig. 2). This converts retargeting from a weighted optimization problem into a set of *closed-form geometric subproblems*. The wrist parameter constrains precise task-space poses (palm and fingers), while the elbow parameter preserves the human arm configuration within the remaining feasible set. Each non-degenerate human pose maps to a *unique robot configuration*. WARP further extends the retargeting algorithm to a full system with constrained alignment for dexterous hands, a task-aware mechanism that selects which body parts to constrain, lazy lower-body control for smooth base-body coordination, and a hierarchical policy that efficiently learns whole-body actions.

We evaluate WARP at both the retargeting and policy-learning levels. On standard retargeting benchmarks, WARP reduces palm-position tracking error by 99.34% over the strongest end-effector baseline, MINK-EF [29], while producing consistent solutions across initializations. In downstream policy learning, identical policies trained on WARP-retargeted data outperform those trained on end-effector-only and optimization-based retargeting by 12 percentage points (71% vs. 59% average task success on DexMimicGen). These results show that WARP produces robot trajectories that are precise, consistent, and learnable: they can be replayed open-loop on the robot, used directly as supervision, and transferred across multiple humanoid embodiments. To our knowledge, WARP is the first system to train a whole-body mobile manipulation policy directly from offline human demonstrations, without human-in-the-loop teleoperation, achieving zero-shot

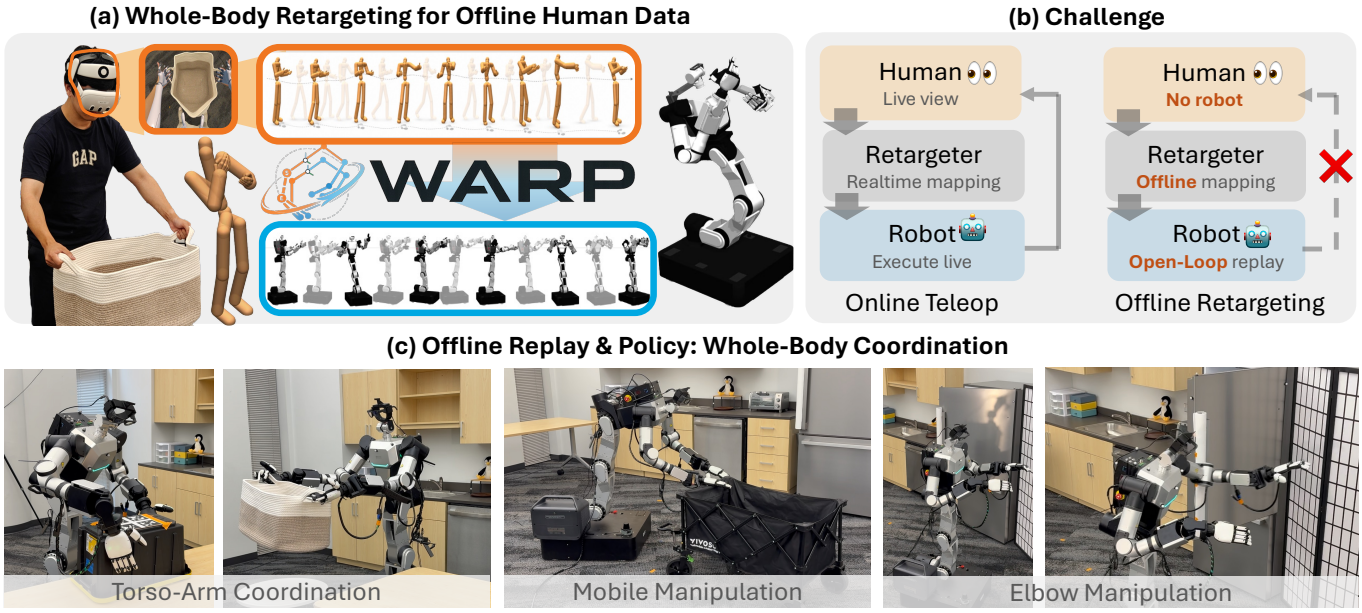


Fig. 1: **Whole-body-Aware Retargeting from human Pose (WARP). (a)** We collect human manipulation data *offline* using VR devices, and WARP retargets this motion into *whole-body robot actions*, producing human-like trajectories directly usable for policy training. **(b)** The central difficulty of the offline setting is the absence of online human correction to close the embodiment gap. With no human in the loop to absorb mismatches, retargeting errors become far less tolerable, and small errors cause replay failure. **(c)** WARP retargets whole-body human motions, creating versatile whole-body coordination skills for manipulation.

transfer on a tractable subset of real-world tasks.

II. Related Work

Behavior Cloning for Mobile Manipulation. Behavior cloning (BC) is the dominant paradigm for visuomotor robot learning: collect expert demonstrations, fit a policy, and deploy it on the target robot. These demonstrations are overwhelmingly gathered by teleoperation— leader–follower rigs (GELLO, Mobile ALOHA), VR systems (Open-TeleVision, Bunny-VisionPro), and hand-held grippers (UMI, DexCap) [25, 11, 5, 8, 7, 23]— and feed expressive policies (ACT, Diffusion Policy) and vision–language–action models (OpenVLA, π_0) [30, 6, 15, 3]. But BC couples data collection to deployment: because demonstrations are tied to the embodiment they were collected on, every new robot needs fresh data. This coupling is most costly for mobile manipulation, whose mobile, whole-body platforms make teleoperation far more hardware- and effort-intensive. Recent systems make strong progress on long-horizon household tasks but remain teleoperation-driven [26, 11, 13, 17, 2, 22, 28]: Mobile ALOHA enables low-cost bimanual whole-body collection, while BEHAVIOR Robot Suite, CLONE, and HOMIE improve control for mobile or humanoid data [13, 17, 2]. Such systems improve demonstration quality but stay locked to embodiment-specific teleoperation hardware. We instead replace robot teleoperation with scalable human demonstrations, using a retargeting system that closes the human–robot embodiment gap in the kinematic action.

Human to Robot Motion Retargeting. Retargeting converts human skeleton keypoints into robot joint trajectories. Vision-based teleoperation systems (AnyTeleop, OpenTeleVision) map tracked upper-body motion to robots via embodiment-specific retargeting or inverse kinematics, trading generality against geometric fidelity [21, 5]. At the humanoid scale, retargeting is increasingly coupled with learned dynamics: HOMIE and CLONE target locomotion-aware whole-body execution, while GMR shows retargeting quality strongly affects downstream motion-tracking [2, 17, 1]. Our setting is quasi-static and manipulation-centric, so we align most closely with SEW-Mimic, a closed-form shoulder–elbow–wrist solver for upper-body humanoid teleoperation [16]; its fast, precise upper-body alignment suits learning mobile manipulation from human demonstrations, where closing the kinematic gap matters more than general locomotion.

III. Method

Learning from offline human demonstrations requires precise, feasible robot action data from retargeting. Therefore, a useful retargeter must satisfy two requirements: *precision*—reproducing the human's whole-body posture, accurately—and *consistency*—mapping identical human poses to a single robot configuration so the policy sees a unimodal target. We meet both with **WARP**, a geometry-based retargeter paired with a policy trained on its output. Its core, c-SEW, solves retargeting in closed form over the Shoulder–Elbow–Wrist representation [16], producing a *unique* configuration that exactly matches the human wrist and preserves whole-body pose similarity. A *lazy lower-body controller* extends this guarantee across the full task, and a *hierarchy-masked flow policy* (Sec. I) learns whole-body mobile manipulation directly from the resulting demonstrations.

Offline retargeting for policy learning. Our goal is to convert offline human demonstrations into whole-body robot actions usable as supervision for a behavior-cloning policy, with no operator in the loop. Given a human whole-body trajectory $\tau^h = \{o_t, \mathcal{H}(t)\}_{t=1}^T$—observation o_t and kinematic skeleton $\mathcal{H}$ information, captured by a MoCap or VR device—and a robot whose kinematic $\mathcal{B}$ differs from the human's, an offline retargeter Ret is a per-timestep map from $\mathcal{H}(t)$ to a robot action $q_t^r = Ret(\mathcal{H}(t))$. The retargeted trajectory $\{q_t^r\}_{t=1}^T$ serves as both the proprioceptive state and the supervision for behavior cloning: the policy π_θ maps the environment observation o_t and a state history $q_{t-L:t}^r$ to the next action chunk $q_{t+1:t+H}^r$, trained by $\min_\theta \mathbb{E}_{t,\,\tau^h} \mathcal{L}\big(\pi_\theta(o_t, q_{t-L:t}^r),\ q_{t+1:t+H}^r\big)$.

SEW representation for humanoid kinematics. The Shoulder–Elbow–Wrist (SEW) representation describes each arm by the geometric skeleton $\mathcal{H} = (\mathbf{s}, \mathbf{e}, \mathbf{w}, \mathbf{H}, \mathbf{t})$, where $\mathbf{s}, \mathbf{e}, \mathbf{w}, \mathbf{t} \in \mathbb{R}^3$ are the shoulder, elbow, wrist positions, and palm offset and $\mathbf{H} \in \mathrm{SO}(3)$ is the palm-center orientation, all expressed in the upper-body-centric frame $\mathbf{T}^{\text{hm}}$ constructed from Algorithm 1. Its key abstraction is to separate arm shape from embodiment scale. The human arm configuration is represented by the scale-invariant limb directions $\mathbf{u} = \mathrm{unit}(\mathbf{e} - \mathbf{s}), \mathbf{l} = \mathrm{unit}(\mathbf{w} - \mathbf{e})$.

For a humanoid arm with a spherical shoulder, pin-joint elbow, and spherical wrist, these SEW constraints admit a closed-form inverse-kinematic solution with at most one valid joint configuration [16]. Thus, once a target robot skeleton $(\mathbf{s}^{\text{rb}}, \mathbf{e}^{\text{rb}}, \mathbf{w}^{\text{rb}}, \mathbf{H}^{\text{rb}})$ is specified, the upper-body joint action can be recovered deterministically: $\mathbf{q}^{\text{rb}} = \text{SEW}\left(\mathbf{s}^{\text{rb}}, \mathbf{e}^{\text{rb}}, \mathbf{w}^{\text{rb}}, \mathbf{H}^{\text{rb}}\right)$. The solution consistency of SEW provides a unique supervision signal and benefit policy learning.

However, vanilla SEW-solver [16] aligns limb directions rather than the palm contact point. Define robot geometry as $\ell_{OS}, \ell_{SE}, \ell_{EW}, \mathbf{p}_{\text{WT}}$, read once from the URDF. Here O, S, E, W, T denote the upper-body origin, shoulder, elbow, wrist, and palm/tool point, respectively. ℓ_{OS} is the origin-to-shoulder offset, ℓ_{SE} is the upper-arm length, ℓ_{EW} is the forearm length, and $\mathbf{p}_{\text{WT}} \in \mathbb{R}^3$ is the fixed wrist-to-tool offset expressed in the hand frame. Since the human and robot have different link lengths, copying $\mathbf{u}$, $\mathbf{l}$, and $\mathbf{H}$ generally produces a systematic palm displacement: $\hat{\mathbf{t}}^{\text{rb}} = \mathbf{s}^{\text{rb}} + \ell_{SE}\mathbf{u} + \ell_{EW}\mathbf{l} + \mathbf{H}^{\text{hm}}\mathbf{p}_{\text{WT}},\ \hat{\mathbf{t}}^{\text{rb}} \neq \mathbf{t}$. This is acceptable for online teleoperation, where a human can compensate, but not for offline policy learning, where the retargeted action is the supervision. WARP therefore uses SEW as the deterministic IK backbone, but replaces direction-only matching with constrained skeleton alignment: the robot palm is forced to coincide with the demonstrated palm, while the remaining geometric freedom is used to preserve the human SEW structure.

WARP's formulation. WARP departs from prior retargeters along two design axes. First, it constructs a target robot SEW skeleton $(\mathbf{s}^{\text{rb}}, \mathbf{e}^{\text{rb}}, \mathbf{w}^{\text{rb}}, \mathbf{H}^{\text{rb}})$ before solving for joint angles, rather than optimizing directly in the redundant joint space $\mathbf{q}$. This decouples cross-embodiment skeleton alignment from joint-angle IK and lets the final robot action be recovered by the closed-form SEW solver. Second, WARP enforces palm matching as a hard constraint, rather than trading off end-effector accuracy and pose similarity through weighted objectives.

Denote the robot geometry defined previously by $\mathcal{B} = (\ell_{OS}, \ell_{SE}, \ell_{EW}, \mathbf{p}_{\text{WT}})$. For each arm, WARP computes

$$\left(\mathbf{T}^{\text{hm}\leftarrow\text{rb}}, \mathbf{s}^{\text{rb}}, \mathbf{e}^{\text{rb}}, \mathbf{w}^{\text{rb}}, \mathbf{H}^{\text{rb}}\right)_{(\text{L},\text{R})} = \text{WARP}\big(\mathcal{H}_{(\text{L},\text{R})}, \mathcal{B}\big), \quad (1)$$

satisfying $\left(\mathbf{w}^{\text{rb}} + \mathbf{H}^{\text{rb}}\mathbf{p}_{\text{WT}}\right)_{(\text{L},\text{R})} = \left(\left[\mathbf{T}^{\text{hm}\leftarrow\text{rb}}\right]^{-1}\mathbf{t}\right)_{(\text{L},\text{R})}$. The constraint states that the robot palm must coincide exactly with the human palm in the aligned robot upper-body frame. The remaining degrees of freedom are resolved analytically to preserve the human SEW structure, as described next. WARP solves for the desired robot SEW representation in two stages: 1) adaptive offset that solves for the optimal robot torso placement, which accounts for major link length difference, 2) per-arm palm alignment that matches human and robot palm pose while solving for robot elbow nullspace use human elbow angle.

Adaptive offset. We first measure how far the robot palm drifts from the human palm when robot aligns its links in human arm directions $\mathbf{u}^{\text{hm}}$, $\mathbf{l}^{\text{hm}}$. Define the SEW forward kinematics for the robot palm: $\hat{\mathbf{t}}^{\text{rb}} = \mathbf{s}^{\text{rb}} + \mathbf{u}^{\text{hm}}\,\ell_{SE} + \mathbf{l}^{\text{hm}}\,\ell_{EW} + \mathbf{H}^{\text{hm}}\,\mathbf{p}_{\text{WT}}$, where $\mathbf{s}^{\text{rb}}$ is the robot shoulder in the upper-body frame and $\mathbf{p}_{\text{WT}} \in \mathbb{R}^3$ is the robot wrist-to-palm offset with $\|\mathbf{p}_{\text{WT}}\| = \ell_{WT}$, read from the URDF. The human palm $\mathbf{t}$ is given directly from the body-centric frame extraction (Sec. III). The two-arm centroid offset that nullifies the net palm displacement is

$$\mathbf{p}_{\text{offset}} = \tfrac{1}{2}\big(\mathbf{t}_{\text{L}} + \mathbf{t}_{\text{R}}\big) - \tfrac{1}{2}\big(\hat{\mathbf{t}}_{\text{L}}^{\text{rb}} + \hat{\mathbf{t}}_{\text{R}}^{\text{rb}}\big). \quad (2)$$

Shifting the robot upper-body origin by $\mathbf{p}_{\text{offset}}$ as shown in Fig. 2 (a) aligns the palm centroids exactly: $\mathbf{T}^{\text{hm}\leftarrow\text{rb}} \leftarrow (\mathbf{p}_{\text{offset}}, \mathbf{I}_3)$

Per-arm palm alignment. With $\mathbf{p}_{\text{offset}}$ applied, each arm is solved independently. Hand orientation transfers directly, $\mathbf{H}^{\text{rb}} = \mathbf{H}^{\text{hm}}$, and the desired robot wrist position is recovered from the human palm target $\mathbf{t}$, transformed to robot frame using $\left[\mathbf{T}^{\text{hm}\leftarrow\text{rb}}\right]^{-1}$: $\mathbf{w}^{\text{rb}} = \left[\mathbf{T}^{\text{hm}\leftarrow\text{rb}}\right]^{-1}\mathbf{t} - \mathbf{H}^{\text{rb}}\,\mathbf{p}_{\text{WT}} + \mathbf{p}_{\text{offset}}$.

It remains to find the robot elbow $\mathbf{e}^{\text{rb}}$ consistent with $\mathbf{s}^{\text{rb}}$, $\mathbf{w}^{\text{rb}}$, and link lengths ℓ_{SE}, ℓ_{EW} as shown in Fig. 2 (b). We preserve the human elbow configuration by extracting the SEW elbow angle ψ which allows us to transfer the unique elbow half-plane normal parametrization $\hat{\mathbf{n}}$ from the human skeleton to robot frame via the stereographic forward and inverse kinematics of Stereo-sew [9]: $\psi \leftarrow$ Stereo-sew.FwdKin$\big(\mathbf{s}, \mathbf{e}, \mathbf{w}\big)$, $\hat{\mathbf{n}} \leftarrow$ Stereo-sew.InvKin$\big(\mathbf{s}^{\text{rb}}, \mathbf{w}^{\text{rb}}, \psi\big)$.

Let $\hat{\mathbf{e}}_{SW} = \mathrm{unit}(\mathbf{w}^{\text{rb}} - \mathbf{s}^{\text{rb}})$. We apply SP3 from [10] to find the rotation θ_{SEW} about $\hat{\mathbf{n}}$ that places the tip of the rotated upper-arm vector at distance ℓ_{EW} from the desired wrist: $\theta_{SEW} = \text{SP3}\big(\ell_{SE}\,\hat{\mathbf{e}}_{SW},\ \mathbf{w}^{\text{rb}} - \mathbf{s}^{\text{rb}},\ \hat{\mathbf{n}},\ \ell_{EW}\big)$. Here, SP3 solves for the direction of the robot upper arm ℓ_{SE}. Among the at most two solutions we select $\theta_{SEW} > 0$, the branch consistent with the human half-plane encoded by $\hat{\mathbf{n}}$. The robot elbow position is then

$$\mathbf{e}^{\text{rb}} = \mathbf{s}^{\text{rb}} + \mathcal{R}\big(\hat{\mathbf{n}},\, \theta_{SEW}\big)\,\big(\ell_{SE}\,\hat{\mathbf{e}}_{SW}\big), \quad (3)$$

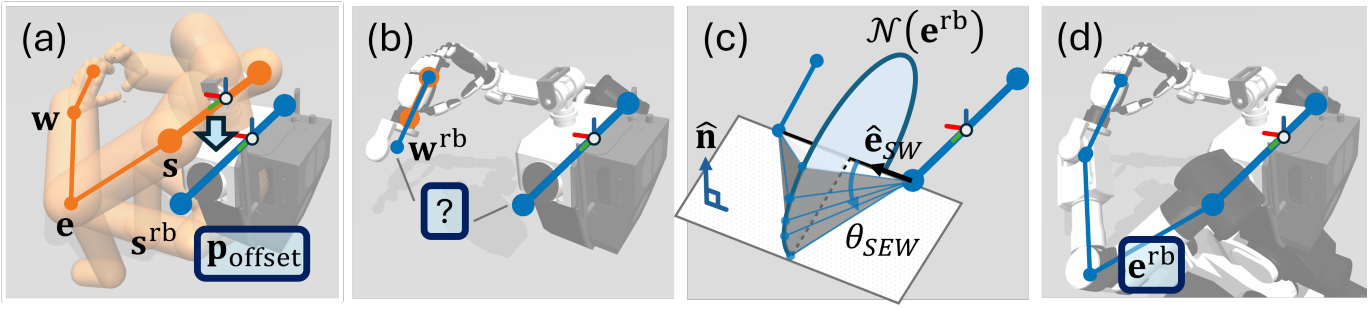


Fig. 2: Offline retargeting with WARP. (a) Finding the optimal robot torso placement using Adaptive Offset (b) After aligning robot palm to human, robot wrist position can be solved (c) Prioritizing EEF alignment leaves the elbow configuration underconstrained. (d) Given fixed robot wrist and shoulder, we identify the elbow nullspace. Using Stereo-sew [9], we find a unique plane intersecting the nullspace circle. (d) The solved robot SEW representation.

where $\mathcal{R}(\hat{\mathbf{n}}, \theta)$ denotes rotation by θ about $\hat{\mathbf{n}}$. The corrected robot skeleton $(\mathbf{s}^{\text{rb}}, \mathbf{e}^{\text{rb}}, \mathbf{w}^{\text{rb}}, \mathbf{H}^{\text{rb}})$ is passed to SEW-Mimic to recover joint angles $\mathbf{q}^{\text{rb}}$ in closed form using SEW-Mimic (Fig. 2 (d)); no iterative solver is invoked at any stage. The 6-DoF torso pose $\mathbf{T}^{\text{torso}} = \mathbf{T}^{\text{hm}} \cdot \mathbf{T}^{\text{hm}\leftarrow\text{rb}}$ is solved using a closed-form IK solver ik-geo [10] in which we always choose the knee-forward solution.

Properties. WARP provides: **(i)** a *unique* solution for any non-degenerate human arm pose through the combination of Stereo-sew, SP3, ik-geo; **(ii)** *exact* palm-position matching to the human palm by construction (Eq. (1)); **(iii)** *whole-body orientation similarity* maximized subject to the EEF constraint via optimal robot placement of Adaptive Offset and the preserved SEW angle ψ; and **(iv)** *closed-form* execution at microsecond speed with no per-robot rescaling or calibration of SEW-Mimic. These properties jointly satisfy the precision and consistency requirements of Sec. III.

Lazy Mobile-Base Tracking. A whole-body target derived from human data implies a base pose $\mathbf{q}_d \in SE(2)$, but tracking it directly couples every small upper-body shift into wheel motion, and the base's high inertia and limited bandwidth turn those corrections into lag and overshoot that destabilize manipulation. Our key insight is that the 6-DoF torso can absorb most small upper-body adjustments, so the base should move only for genuine relocation. WARP therefore maintains a *lazy* base target $\mathbf{q}_b$ instead of tracking $\mathbf{q}_d$ directly, and solves torso and arm joints around that target. In effect, the torso handles fine adjustment while the base handles coarse repositioning, improving manipulation stability without sacrificing mobility. Calculation of $\mathbf{q}_b$ are given in Sec. G.

Data Collection and Robot System We collect demonstrations using a single Meta Quest headset, without external motion-capture rigs or robots in the loop. Unlike traditional teleoperation and UMI-style interfaces, which compress a demonstration down to the end-effector's spatial pose, we capture the operator's whole-body motion. The system logs the operator's root locomotion alongside a 6-DoF upper-body kinematic tree with full hand articulation in a fixed world frame at 60 Hz (Fig. 1). We deploy the RB-Y1 humanoid (Fig. 3), featuring a holonomic base, 6-DoF torso, two 7-DoF arms, and 12-DoF five-fingered XHands. The robot is commanded via joint-impedance control at 100 Hz. To isolate our kinematic and policy contributions from perception and odometry drift, we track manipulated objects using AprilTags (via head-mounted Project Aria Gen 2 glasses [20]) and localize the robot base using a Vicon system, providing precise localization required for replay.

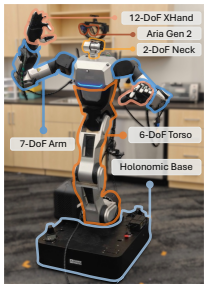


Fig. 3: Our whole-body mobile manipulation platform.

IV. Experiments

Our experiments validate that WARP's retargeting solution is precise, consistent, and whole-body human-like, and that these properties make the resulting robot data more learnable–improving whole-body mobile-manipulation policies. We evaluate the following hypotheses **H1 Retargeting quality.** WARP produces higher-quality whole-body motion than end-effector-only and generic whole-body retargeting baselines, as measured by tracking accuracy, hardware feasibility, solution consistency, and solving speed on our mobile manipulation platform (Fig. 3). **H2 Downstream learnability.** The motion properties produced by WARP translate into measurably better policy learning. Identical policies trained on WARP-retargeted data outperform those trained on baseline-retargeted data in success rate and rollout consistency. **H3 Necessity of whole-body retargeting.** For tasks whose success depends on null-space configuration (torso, elbow, or base placement), end-effector-only retargeting is fundamentally insufficient regardless of human data quality and quantity, while WARP succeeds.

A. Evaluation on Retargeting Motion Quality

Setup. We evaluate retargeted motion data quality using a high-quality human motion dataset BONES-SEED [4]. It preserves per-actor body proportions and captures realistic human motion geometry. From it, we sample 514 manipulation demonstrations via language search over the provided human motion descriptions Sec. L1. We compare WARP against two retargeting baselines spanning the dominant design points

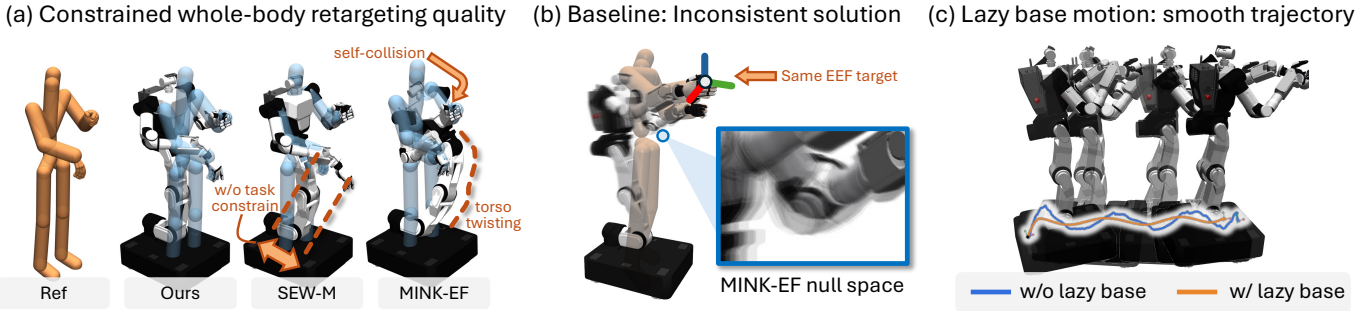
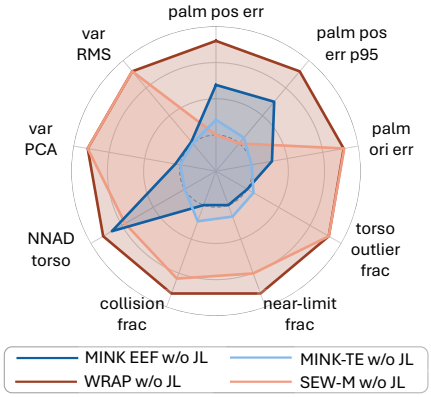


Fig. 4: WARP **is precise, consistent, and smooth.** **(a)** WARP matches both the task constraint (end-effector pose) and the human body pose; baselines satisfy only one—SEW-M loses the end-effector, MINK-EF self-collides and twists the torso. **(b)** Under a perturbed initial guess with the end-effector target fixed, MINK's optimization is inconsistent, while WARP's closed-form solution is identical every time. **(c)** A responsive upper body compensates for the low-frequency *lazy base*, also smoothing the base trajectory.



Method	JL	Tracking error ↓			Feasibility ↓			Solver variation ↓		
		Palm mm	P95 mm	Ori. deg	Torso frac.	Limit frac.	Coll. frac.	NNAD –	PCA eig.	RMS deg
SEW-M	off	178.979	201.056	**7.89e-6**	**0.000**	0.0126	0.243	0.488	1.20e-25	6.83e-14
MINK-EF	off	0.701	1.853	0.0107	0.625	0.1610	0.977	0.368	173.49	3.117
MINK-TE	off	18.557	73.980	0.157	0.027	0.0852	0.640	1.026	1119.11	6.106
WARP	off	**0.0046**	**0.046**	8.74e-6	**0.000**	**0.0047**	0.163	0.289	1.14e-25	6.66e-14
SEW-M	on	215.641	272.842	6.304	0.162	0.0131	0.084	0.442	1.09e-25	6.56e-14
MINK-EF	on	0.751	2.478	0.0112	0.611	0.1247	0.222	0.354	227.29	3.169
MINK-TE	on	19.492	69.345	0.188	0.049	0.0420	0.478	0.454	613.47	4.964
WARP	on	24.048	82.036	3.259	0.130	0.0060	**0.017**	**0.266**	**1.06e-25**	**6.46e-14**

Fig. 5: Simulation retargeting results. Left: radar visualization of retargeting feasibility and motion-quality diagnostics. Right: quantitative results for the highlighted variants. All metrics are lower-is-better. Best results are shown in **bold**; second-best results are underlined.

in the literature: **MINK** [29] (optimization-based IK, end-effector-only) MINK-EF, **body constrained MINK** MINK-TE (end-effector, torso orientation, and elbow swivel angle introduced in Eq. (3)), and **SEW-M** [16] (SEW-Mimic, unconstrained).

Metrics. We evaluate retargeting along three axes, all lower-is-better unless noted. **Tracking accuracy** captures fidelity to the target human pose via the mean and 95th-percentile palm position error, and the palm orientation error. **Hardware feasibility** checks whether a kinematical solution is safe on the real robot via the torso outlier rate, the joints near-limit rate, and the self-collision rate. **Solution consistency**, which matters for policy learning, is measured by NNAD, PCA, and the RMS standard deviation of joint angles. We give all metric formulas in Appendix J0a.

WARP produces accurate, feasible, and consistent retargeting solutions. It achieves the lowest tracking error among all JL-off methods, as shown in Fig. 5: even without auxiliary constraints or regularization, it reduces palm position error by more than 150× over MINK-EF and drives hand-orientation error to machine precision, both direct consequences of the closed-form SEW solver, which enforces exact tracking by construction rather than through a weighted cost. (Its JL-on error is larger, but mainly because aggressive motions in the BONES-SEED proportional dataset push some targets outside the robot's reachable workspace under a human-like posture, where less human-like methods like MINK can approach them more closely.) WARP also produces the most feasible motion, with the lowest joint-limit and self-collision fractions: a humanoid's large null space admits many task-satisfying but physically infeasible configurations that exceed joint limits or self-collide, so a retargeter should encode the human pose prior to suppress them, and WARP's human-like solutions do exactly that, acting as a self-collision-avoidance prior that baselines capture less well (Fig. 4). It is likewise the most consistent across variants, with the best NNAD, PCA, and RMS solver-variation scores by orders of magnitude; unlike optimization-based approaches, its solutions are deterministic and initial-condition-independent, so the policy learns from consistent, non-conflicting targets rather than noisy, seed-dependent ones. Finally, WARP solves roughly 30× faster than iterative optimization-based retargeters—converting BONES-SEED takes about an hour on one CPU versus a full day for the baseline—leaving headroom for online use.

B. Simulation Evaluation

a) Setup.: DexMimicGen [14] synthesizes large dexterous-manipulation datasets from a small set of seed teleoperation demonstrations through task-structured replay with randomized object resets. We use 200 synthesized GR1 demonstrations per task across three bimanual benchmarks—*coffee*, *pouring*, and *can-sort*. For each demonstration, we retarget the motion to the RB-Y1 humanoid with Fourier

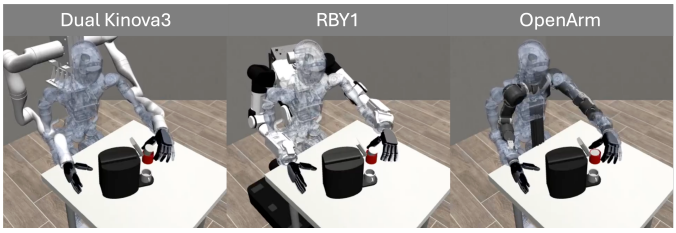


Fig. 6: WARP retargets one robot motion to different robot embodiments.

	can_sort		*pouring*		*coffee*		*average*	
Method	replay	policy	replay	policy	replay	policy	replay	policy
MINK	**99.5%**	94%	88.5%	74%	50.5%	8%	79.5%	59%
WARP	98.5%	**100%**	**90.5%**	**78%**	**51.0%**	**34%**	**80.0%**	**71%**

TABLE I: Replay and policy rollout result (%) on DexMimicGen tasks. WARP achieves 12% higher success rate for policy.

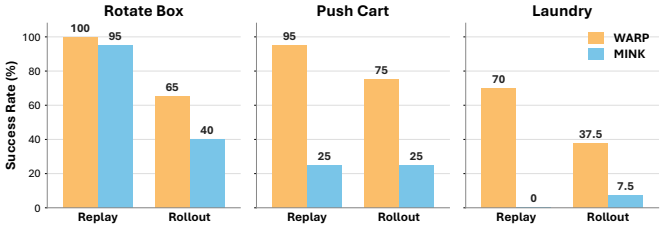


Fig. 7: Real-world evaluation of retargeted data replay and policy rollout (10 trials each).

hands using both WARP and MINK, then filter the results through MuJoCo replay with stochastic retries. Finally, we train one behavior-cloning policy for each task-retargeter pair using the demonstrations that succeed under both retargeters, i.e., the WARP–MINK success intersection.

WARP retarget data leads to higher task success. At replay of DexMimicGen tasks, WARP and MINK reach comparable success rates—open-loop execution of either retargeter's output completes the task at similar rates—but the gap opens during policy training: policies trained on WARP data achieve substantially higher task-execution success than those trained on MINK data, with the margin widest on fine-grained manipulation such as the coffee task. Replay parity therefore understates the difference between retargeters. The feasibility and solution-consistency properties do not materially change what is replayable for robot data, but they directly shape what a policy can learn from the resulting data—so retargeting nuances invisible at replay become decisive downstream. We further shows that WARP can transfer robot data to different embodiments as shown in Fig. 6.

C. Real-World Evaluation

We collect 50 human demonstrations for each task and train a policy with it (Sec. I). **Pick-up-laundry.** The operator lifts a laundry basket by both handles, moves it to an adjacent table, and sets it down. The task stresses bimanual wrist control: each handle must be gripped at a precise orientation and that orientation re-achieved to release, so palm position alone is insufficient. **Push-cart.** The operator pushes a small cart forward by $0.5\,\mathrm{m}$ with both hands. The task tests base–arm coordination under contact: success depends not just on where the hands go but on whether the body clears the cart. **Rotate-box.** The operator grips opposite sides of a box and rotates it 90° clockwise. The task stresses torso–arm coordination, since the robot must twist its upper body to place both end-effectors at the required orientation. **Fridge-door-closing (elbow mode).** The operator closes a fridge door with both hands occupied, pushing it shut with the left elbow—contact with a body link other than the hands, outside any end-effector-only formulation. Without proper observation, we train no policy and evaluate replay, where WARP succeeds in 90% of trials.

Finding Summary. Across all four tasks, WARP outperforms the end-effector-centric MINK baseline (Fig. 7). By coordinating torso and arm as the demonstrator did, WARP keeps palm contact and recoverable posture, whereas MINK drives the upper body into extreme configurations feasible only at open-loop replay (rotate-box, pick-up-laundry). Its whole-body objective exploits the full kinematic chain: WARP supports manipulation with arbitrary links such as the elbow (fridge-door-closing) and preserves the implicit base–arm constraints that end-effector- or keypoint-based retargeting discards by construction (push-cart, verifying H3). Its closed-form SEW solver (Finding 1) tracks the hand frame $\mathbf{H}$ to machine precision, guaranteeing the wrist alignment that grasp and release demand—where MINK's accumulated orientation errors become decisive (pick-up-laundry). The result is more learnable data: WARP replays the fridge task in 90% of trials and, even at comparable replay rates, trains stronger policies (65% vs. 40% on rotate-box; Fig. 7). Motion quality, not replay completion, governs downstream success (Fig. 9).

V. Conclusion

We present WARP, an offline framework of human-to-robot retargeting and policy-learning for whole-body mobile manipulation by leveraging SEW representation between human and robot pose retargeting. To our knowledge, WARP is the first system to train a whole-body mobile manipulation policy directly from offline human demonstrations, without teleoperation.This supports further research on policy learning of high-quality human pose data for zero-shot transfer on a tractable subset of real-world tasks.

Limitation Image observation is lacking for our policy training, limiting the tasks we could try. We would try image policy in the next step and investigate how WARP can help in visual-motor policy from offline human demonstration.

References

[1] Joao Pedro Araujo, Yanjie Ze, Pei Xu, Jiajun Wu, and C. Karen Liu. Retargeting matters: General motion retargeting for humanoid motion tracking, 2025. URL https://arxiv.org/abs/2510.02252.

[2] Qingwei Ben, Feiyu Jia, Jia Zeng, Junting Dong, Dahua Lin, and Jiangmiao Pang. HOMIE: Humanoid loco-

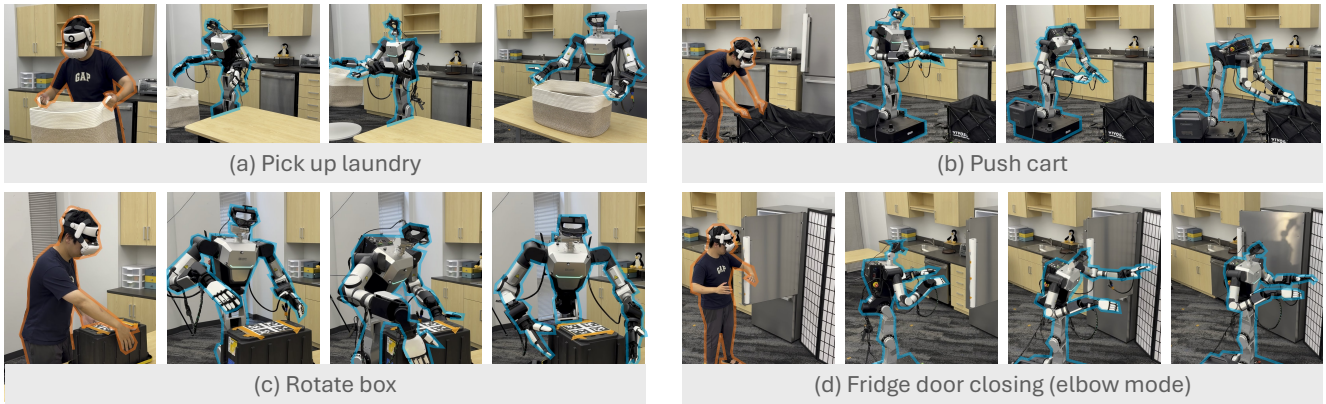


Fig. 8: **Human demonstrations** and **robot executions** across four real-world tasks.

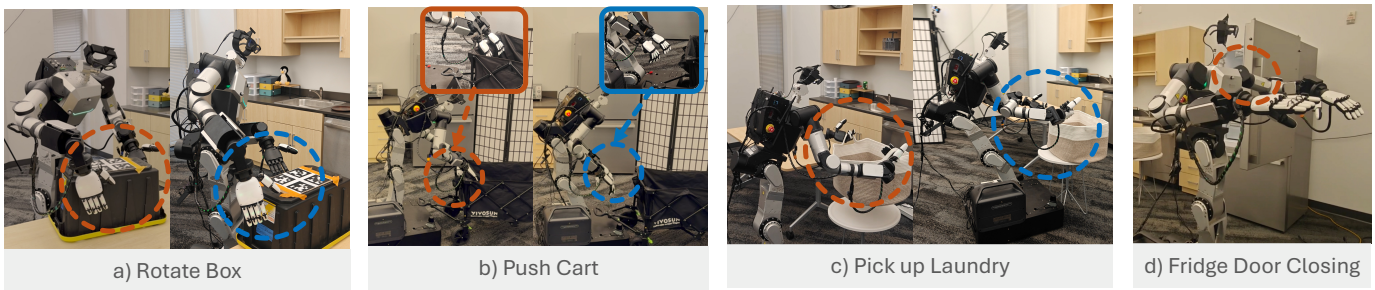


Fig. 9: Comparison with **WARP** and **MINK** on our tasks. WARP resolves constraints across the whole kinematic chain, preserving palm pose and human-like posture—a motion-quality advantage driving policy success. MINK drives the torso and base into extreme or colliding configurations: a) over-rotating, b) hitting the cart, and c) accumulating wrist-orientation errors that break grasps. d) shows that WARP can also treat the elbow as a task-space objective.

manipulation with isomorphic exoskeleton cockpit, 2025. URL https://arxiv.org/abs/2502.13013.

[3] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2024.

[4] Bones Studio. BONES-SEED: Skeletal everyday embodiment dataset. https://huggingface.co/datasets/bones-studio/seed, 2026. Hugging Face dataset. Accessed: 2026-05-10.

[5] Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: Teleoperation with immersive active visual feedback, 2024. URL https://arxiv.org/abs/2407.01512.

[6] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.

[7] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Robotics: Science and Systems (RSS)*, 2024.

[8] Runyu Ding, Yuzhe Qin, Jiyue Zhu, Chengzhe Jia, Shiqi Yang, Ruihan Yang, Xiaojuan Qi, and Xiaolong Wang. Bunny-VisionPro: Real-time bimanual dexterous teleoperation for imitation learning, 2024.

[9] Alexander J Elias and John T Wen. Redundancy parameterization and inverse kinematics of 7-dof revolute manipulators. *Mechanism and Machine Theory*, 204: 105824, 2024.

[10] Alexander J Elias and John T Wen. Ik-geo: Unified robot inverse kinematics using subproblem decomposition. *Mechanism and Machine Theory*, 209:105971, 2025.

[11] Zhe Fu, Tony Z. Zhao, and Chelsea Finn. Mobile ALOHA: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. In *Proceedings of The 8th Conference on Robot Learning*, 2024.

[12] Irmak Guzey, Haozhi Qi, Julen Urain, Changhao Wang, Jessica Yin, Krishna Bodduluri, Mike Maroje Lambeta, Lerrel Pinto, Akshara Rai, Jitendra Malik, Tingfan Wu, Akash Sharma, and Homanga Bharadhwaj. Dexterity from smart lenses: Multi-fingered robot manipulation with in-the-wild human demonstrations, 2025. URL https://arxiv.org/abs/2511.16661.

[13] Yunfan Jiang, Ruohan Zhang, Josiah Wong, Chen Wang, Yanjie Ze, Hang Yin, Cem Gokmen, Shuran Song, Jiajun Wu, and Li Fei-Fei. BEHAVIOR robot suite: Streamlining real-world whole-body manipulation for everyday household activities, 2025. URL https://arxiv.org/abs/

2503.05652.

[14] Zhenyu Jiang, Yuqi Xie, Kevin Lin, Zhenjia Xu, Weikang Wan, Ajay Mandlekar, Linxi Fan, and Yuke Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[15] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024.

[16] Chuizheng Kong, Yunho Cho, Wonsuhk Jung, Idris Wibowo, Parth Shinde, Sundhar Vinodh-Sangeetha, Long Kiu Chung, Zhenyang Chen, Andrew Mattei, Advaith Nidumukkala, Alexander Elias, Danfei Xu, Taylor Higgins, and Shreyas Kousik. A closed-form geometric retargeting solver for upper body humanoid robot teleoperation, 2026. URL https://arxiv.org/abs/2602.01632.

[17] Yixuan Li, Yutang Lin, Jieming Cui, Tengyu Liu, Wei Liang, Yixin Zhu, and Siyuan Huang. CLONE: Closed-loop whole-body humanoid teleoperation for long-horizon tasks, 2025. URL https://arxiv.org/abs/2506.08931.

[18] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.02747.

[19] Vincent Liu, Ademi Adeniji, Haotian Zhan, Raunaq Bhirangi, Pieter Abbeel, and Lerrel Pinto. Egozero: Robot learning from smart glasses, 2025. URL https://arxiv.org/abs/2505.20290.

[20] Project Aria Team. Aria Gen 2: An advanced research device for egocentric ai research. https://www.projectaria.com/ariagen2devicepaper, 2025. Accessed: 2026-05-29.

[21] Yuzhe Qin, Wei Yang, Binghao Huang, Karl Van Wyk, Hao Su, Xiaolong Wang, Yu-Wei Chao, and Dieter Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system, 2023. URL https://arxiv.org/abs/2307.04577.

[22] Priya Sundaresan, Rhea Malhotra, Phillip Miao, Jingyun Yang, Jimmy Wu, Hengyuan Hu, Rika Antonova, Francis Engelmann, Dorsa Sadigh, and Jeannette Bohg. Homer: Learning in-the-wild mobile manipulation via hybrid imitation and whole-body control, 2025. URL https://arxiv.org/abs/2506.01185.

[23] 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. In *Robotics: Science and Systems (RSS)*, 2024.

[24] Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2409.20537.

[25] Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. GELLO: A general, low-cost, and intuitive teleoperation framework for robot manipulators. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[26] Haoyu Xiong, Russell Mendonca, Kenneth Shaw, and Deepak Pathak. Adaptive mobile manipulation for articulated objects in the open world, 2024. URL https://arxiv.org/abs/2401.14403.

[27] Xiaomeng Xu, Jisang Park, Han Zhang, Eric Cousineau, Aditya Bhat, Jose Barreiros, Dian Wang, Jeannette Bohg, and Shuran Song. HoMMI: Learning whole-body mobile manipulation from human demonstrations. *arXiv preprint arXiv:2603.03243*, 2026.

[28] Jingyun Yang, Isabella Huang, Brandon Vu, Max Bajracharya, Rika Antonova, and Jeannette Bohg. Mobi-π: Mobilizing your robot learning policy. *arXiv preprint arXiv:2505.23692*, 2025.

[29] Kevin Zakka. Mink: Python inverse kinematics for robotic systems. https://github.com/kevinzakka/mink, 2024. Software library.

[30] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems (RSS)*, 2023.

APPENDIX
SUPPLEMENTARY MATERIAL

Existing whole-body retargeters—e.g., GMR [1], built on MINK [29]—make the opposite choice on both axes: they optimize in joint space and they track the wrist as a soft objective rather than a hard constraint. Concretely, they pose retargeting as a weighted multi-task velocity-level IK,

$$\dot{\mathbf{q}}^* = \arg\min_{\dot{\mathbf{q}}} \sum_i w_i \left\| J_i \, \dot{\mathbf{q}} - \dot{x}_i^* \right\|^2 + w_{\text{posture}} \left\| \dot{\mathbf{q}} - \dot{\mathbf{q}}_{\text{posture}} \right\|^2, \tag{4}$$

where each task term i is a soft whole-body objective—wrist pose, torso orientation, elbow-swivel angle, or head orientation introduced in Sec. A—and the final term is posture regularization. Wrist and palm tracking are thus one weighted term among many, never enforced exactly, and hand and finger articulation fall outside the formulation entirely.

This conflates cross-embodiment skeleton matching with joint-angle IK, producing three failure modes that violate the replayability and consistency requirements above. **(1) Inconsistency:** the velocity update descends to whichever null-space branch is nearest the seed $\mathbf{q}_0$, so different seeds satisfy the same task yet settle in disjoint regions of joint space, giving a multi-modal training target that confuses policy learning. **(2) Imprecision:** the task weights couple wrist accuracy to posture—raising the posture weight for whole-body imitation degrades wrist tracking and breaks replayability, while lowering it drives the robot toward unnatural or unsafe configurations. **(3) Kinematic Embodiment Gap:** difference between robot-skeleton keypoints and unscaled human keypoints systematically bias the retargeted poses, and retargeting methods are sensitive to how reference points are mapped [1].

WARP removes (1) by construction, returning one closed-form solution per pose; (2) by promoting palm matching to a hard constraint; and (3) by operating on scale-invariant SEW geometry rather than absolute keypoint coordinates.

A. MINK Baseline Variants

We mainly evaluate two variants of the MINK-based retargeting baseline to separate the effect of end-effector tracking from additional whole-body imitation terms. Both variants use the same MINK solver and formulate retargeting as a weighted velocity-level IK problem over the robot body joints. At each iteration, the solver minimizes the active task-space residuals and integrates the resulting joint velocity to update the robot configuration. **MINK's formulation.** Both MINK variants use the same weighted velocity-level IK solver. The end-effector objective tracks the pose of both robot palms, including their 3D positions and orientations. This encourages the robot hands to follow the target palm trajectories extracted from the human motion. The torso objective tracks the orientation of the robot torso against the target upper-body orientation, encouraging whole-body alignment beyond the hands. The elbow objective matches the arm swivel configuration using a scalar SEW angle, rather than directly tracking the 3D elbow position. This provides a compact way to encourage similar arm posture while avoiding over-constraining the elbow location. The **MINK-EF** variant only uses the end-effector objective and therefore focuses on accurate palm tracking. The **MINK-TE** variant additionally enables the torso and elbow objectives, aiming to imitate more of the human whole-body motion. This comparison allows us to isolate whether adding whole-body imitation terms improves retargeting quality beyond palm-pose tracking alone.

Task / parameter	MINK-EF	MINK-TE
Palm position cost	1.0	1.0
Palm orientation cost	1.0	1.0
Torso orientation cost	0.0	0.5
Elbow-swivel cost	0.0	0.2

TABLE II: MINK baseline variants used in our paper visualization. MINK-EF tracks only the palm pose, while MINK-TE additionally uses soft torso-orientation and elbow-swivel objectives.

MINK combines these terms as weighted soft objectives,

$$\dot{\mathbf{q}}^* = \arg\min_{\dot{\mathbf{q}}}; w_{\text{EEF}} \mathcal{C}_{\text{EEF}} + w_T \mathcal{C}_T + w_\psi \mathcal{C}_\psi + w_{\text{posture}} \mathcal{C}_{\text{posture}}$$

Finger motion is not optimized by MINK and is instead solved by the analytical XHand IK module.

a) MINK-EF.: MINK-EF is the end-effector-only variant. It activates the left and right palm-pose objectives, with palm position and orientation costs both set to 1.0. MINK-EF only asks the robot to match the target palm poses, while the remaining redundant degrees of freedom are resolved implicitly by the optimizer, posture regularization, damping, and initialization. This variant tests whether end-effector tracking alone is sufficient for cross-embodiment replay.

b) MINK-TE.: MINK-TE extends MINK-EF by adding soft torso and elbow imitation terms. In addition to the same left and right palm-pose objectives used in MINK-EF, MINK-TE activates a torso orientation objective with cost 0.5 and a scalar elbow-swivel objective with cost 0.2. The elbow term does not directly match the human elbow position in 3D. Instead, it matches the stereographic SEW swivel angle ψ computed from the shoulder, elbow, and wrist points.

For the paper visualization, we use MINK-EF and MINK-TE as the two MINK baselines. MINK-EF represents a minimal end-effector tracking baseline, while MINK-TE represents a stronger MINK baseline with additional torso and elbow posture cues. Both variants remain weighted soft-objective IK methods: palm tracking, torso tracking, and elbow-swivel matching are balanced by task weights rather than enforced as closed-form geometric constraints.

B. MINK Tasks Tradeoff

MINK exposes posture quality as a soft cost weighted against end-effector (EEF) tracking, so one scalar weight might trade the two off to some extend; WARP fixes posture analytically and exposes no such knob. We make this concrete by sweeping each posture weight on the same demonstrations

and reporting palm (EEF) error, the relevant posture error, and NNAD (action variability) against the tuning-free WARP reference (Fig. 10).

Elbow sweep (panels a–c). At `elbow_angle_cost`$= 0$, MINK ignores arm posture: elbow-angle error reaches $76°$ while palm error stays small (1.2 mm). Raising the weight drives elbow error to a $\sim 7°$ minimum near reference, but degrades both other axes: palm error climbs to a 4.9 mm peak at 0.2, and torso NNAD rises monotonically from 0.2 to 2.5, crossing the WARP level (0.3) at ~ 0.05. No weight recovers WARP's simultaneous near-zero elbow error, palm error, and torso variance.

Torso sweep (panels d–f). Leaving the torso redundancy unregularized (`torso_orientation_cost`$= 0$) is worst on every axis: $44°$ torso error, 11.7 mm palm error, and 4.25 arm NNAD. A small weight (0.25) collapses all three, but only torso error keeps improving (to $\sim 3°$); palm error plateaus near 4.7 mm and arm NNAD near 1.3, both far above the WARP references (~ 0 mm and 0.7).

Across both sweeps, every setting that lowers posture error degrades palm accuracy and inflates NNAD in the other body part, and no setting matches SEW on all three at once. WARP attains the best value on each metric simultaneously with no per-task tuning: posture quality follows structurally from the analytic solution rather than from a regularizer. For offline retargeting, this means MINK forces a per-task choice between accurate and consistent supervision, whereas WARP supplies both unconditionally.

C. Body-Centric Frame from Keypoints

We express the SEW skeleton $(\mathbf{s}, \mathbf{e}, \mathbf{w}, \mathbf{H}, \mathbf{t})$ and every tracked pose in an upper-body-centric frame, so the representation depends only on relative geometry rather than on where the operator stands or which way they face. The same routine builds the human frame $\mathbf{T}^{\text{hm}}$ and the robot frame $\mathbf{T}^{\text{rb}}$ from each embodiment's shoulders, giving one shared convention in which the two skeletons are directly comparable. Algorithm 1 constructs this frame in closed form at each timestep from three keypoints.

a) Construction.: The frame is anchored at the shoulder midpoint $\mathbf{p} = \frac{1}{2}(\mathbf{s}_{\text{L}} + \mathbf{s}_{\text{R}})$, and its orientation is fixed by two cues. The lateral axis $\mathbf{u}_{\text{y}}$ runs along the shoulder line from the right to the left keypoint. The remaining axis is pinned by the anchor $\mathbf{p}_{\text{anchor}}$: $\mathbf{p} - \mathbf{p}_{\text{anchor}}$ gives a coarse trunk reference, and $\mathbf{u}_{\text{x}} = \text{unit}\big(\mathbf{u}_{\text{y}} \times (\mathbf{p} - \mathbf{p}_{\text{anchor}})\big)$ takes the direction orthogonal to the shoulder line in the shoulder–anchor plane. With the anchor below the shoulders, $\mathbf{u}_{\text{x}}$ points anteriorly—the body's facing direction. The vertical axis $\mathbf{u}_{\text{z}} = \mathbf{u}_{\text{x}} \times \mathbf{u}_{\text{y}}$ completes a right-handed orthonormal basis $\mathbf{R} = [\mathbf{u}_{\text{x}}, \mathbf{u}_{\text{y}}, \mathbf{u}_{\text{z}}]$.

The construction is deliberately asymmetric in what it trusts. Only the shoulder line enters $\mathbf{R}$ exactly; the anchor contributes a single half-plane choice, and its component along $\mathbf{u}_{\text{y}}$ is removed by the cross product, so anchor noise cannot tilt the frame. The sole requirement is that the three keypoints be non-collinear—if $\mathbf{p}_{\text{anchor}}$ fell on the shoulder line the first cross product would vanish—which holds for any natural upper-body posture.

Algorithm 1 Make Frame from Keypoints
$\mathbf{T} \leftarrow \text{MakeFrame}(\mathbf{s}_{\text{L}}, \mathbf{s}_{\text{R}}, \mathbf{p}_{\text{anchor}})$
// Given three non-collinear keypoints in $\mathbb{R}^3$, output a coordinate frame pose in $\text{SE}(3)$ *parameterized by a rotation matrix and translation vector*

Require: Left, right, and anchor keypoints $\mathbf{s}_{\text{L}}, \mathbf{s}_{\text{R}}, \mathbf{p}_{\text{anchor}}$
// Set translation vector between left / right keypoints
$\mathbf{p} \leftarrow \frac{1}{2}(\mathbf{s}_{\text{L}} + \mathbf{s}_{\text{R}})$
// Represent frame orientation as a rotation matrix
$\mathbf{u}_{\text{y}} \leftarrow \text{unit}(\mathbf{s}_{\text{L}} - \mathbf{s}_{\text{R}})$
$\mathbf{u}_{\text{x}} \leftarrow \text{unit}(\mathbf{u}_{\text{y}} \times (\mathbf{p} - \mathbf{p}_{\text{anchor}}))$
$\mathbf{u}_{\text{z}} \leftarrow \mathbf{u}_{\text{x}} \times \mathbf{u}_{\text{y}}$
$\mathbf{R} \leftarrow [\mathbf{u}_{\text{x}}, \mathbf{u}_{\text{y}}, \mathbf{u}_{\text{z}}]$
$\mathbf{T} \leftarrow \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix}$
return homogeneous matrix $\mathbf{T}$

D. Elbow Position as Retargeting Goal

WARP first recovers the full robot skeleton $(\mathbf{s}^{\text{rb}}, \mathbf{e}^{\text{rb}}, \mathbf{w}^{\text{rb}}, \mathbf{H}^{\text{rb}})$ with exact palm matching and the preserved elbow angle ψ; this fixes the robot elbow $\mathbf{e}^{\text{rb}}$, the intersection of the upper-arm and forearm segments. We then align the robot elbow to the human elbow on the ground plane only. Many everyday elbow motions, such as opening or holding a door, depend on the elbow's horizontal placement, not its height, so a 2D projection match captures the relevant constraint without over-constraining the palm-matched arm.

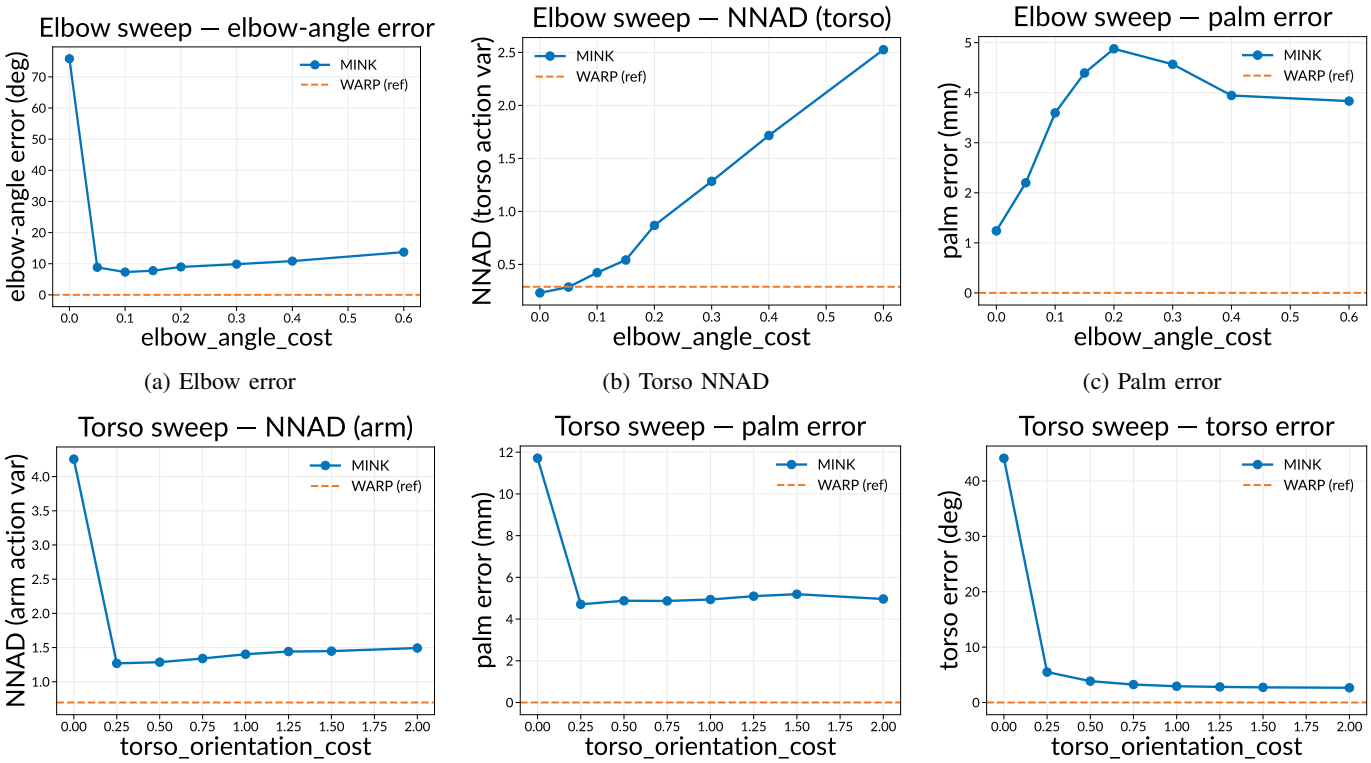
Let $\text{proj} : \mathbb{R}^3 \to \mathbb{R}^2$ project onto the ground plane by dropping the gravity axis. A planar base shift translates every robot elbow projection equally, so, mirroring the adaptive offset of Eq. (2), we take the shift that aligns the two-arm elbow centroids:

$$\mathbf{p}_{\text{elbow}} = \tfrac{1}{2}\big(\text{proj}(\mathbf{e}^{\text{hm}}_{\text{L}}) + \text{proj}(\mathbf{e}^{\text{hm}}_{\text{R}})\big) - \tfrac{1}{2}\big(\text{proj}(\mathbf{e}^{\text{rb}}_{\text{L}}) + \text{proj}(\mathbf{e}^{\text{rb}}_{\text{R}})\big) \tag{5}$$

Applying $\mathbf{p}_{\text{elbow}}$ sets the desired planar base pose $\mathbf{q}_d = (\mathbf{p}_{\text{elbow}}, \theta_d) \in SE(2)$, with yaw θ_d from the human root heading. The 6-DoF torso is then re-solved on $\mathbf{q}_d$ by ik-geo to restore the exact palm constraint of Eq. (1): the base provides coarse horizontal placement and the torso absorbs the residual, so the elbow projection is matched without disturbing palm tracking.

E. Potential Elbow-Position Mode Application Extension

WARP's skeleton-first formulation extends beyond palm-centered retargeting to task-specific priority modes for different body parts. Unlike UMI-style interfaces, which retarget only the 6-DoF end-effector pose, representing the full arm skeleton lets the active target be selected from a set of body-part classes—palm, elbow, torso, or head—which matters when the end-effector alone does not capture the intended behavior. When pushing against or holding a door, for



(a) Elbow error (b) Torso NNAD (c) Palm error

(d) Arm NNAD (e) Palm error (f) Torso error

Fig. 10: **Posture-cost sweeps: MINK (blue) vs. the tuning-free** WARP **reference (orange dashed).** Panels (a)–(c) sweep `elbow_angle_cost`; (d)–(f) sweep `torso_orientation_cost`. No single weight matches SEW on palm accuracy, posture error, and action consistency at once; see Sec. B.

example, the horizontal elbow placement, not the palm pose alone, shapes whether the arm posture is natural and feasible.

As an initial proof of concept, we frame task-target selection as a classification problem: a VLM labels the task-relevant body part from the task instruction or demonstration, and the corresponding mode is activated during offline processing. In elbow-position mode, WARP recovers the full arm skeleton with exact palm matching and preserved elbow angle as before, then adjusts the planar base pose to align the robot elbow projection with the human elbow projection on the ground plane. The hard palm constraint is preserved throughout, so this base adjustment fixes elbow placement without disturbing palm tracking.

To avoid discontinuities when the active label changes along a trajectory, we linearly blend adjacent phases in robot joint space and base $SE(2)$ position, yielding continuous transitions while preserving the per-phase target. Elbow-position mode thus illustrates WARP as an application-level extension: rather than tuning weighted IK costs, we switch the geometric target by task context. We test only elbow-position mode here as a proof of concept; a systematic evaluation of VLM-driven mode selection across tasks and body parts remains future work.

F. Subproblems used in WARP

WARP's geometric solver is built on the geometric *subproblems* [10]: canonical primitives that map a small set of vectors, axes, and scalars to the joint angle(s) satisfying a fixed geometric relation, each solved in closed form with ATAN2 and elementary operations so the result is exact and singularity-robust. The cross-embodiment elbow placement of Eq. (3) uses one of them, SP3.

a) Subproblem 3 (circle and sphere).: Given vectors $p_1, p_2 \in \mathbb{R}^3$, a unit axis $k \in \mathbb{R}^3$, and a nonnegative scalar d, SP3 returns the angle θ that minimizes $\big|\, \|R(k,\theta)\, p_1 - p_2\| - d \,\big|$. Geometrically, $R(k,\theta)\, p_1$ sweeps a circle about k, and the solutions are its intersections with the sphere of radius d centered at p_2: at most two exact angles, with a continuous least-squares angle when the circle and sphere do not meet.

b) Use in WARP.: After per-arm palm alignment fixes the robot shoulder $\mathbf{s}^{\text{rb}}$ and wrist $\mathbf{w}^{\text{rb}}$ and transfers the elbow half-plane normal $\hat{\mathbf{n}}$ from the human, the only remaining unknown is the elbow $\mathbf{e}^{\text{rb}}$, which must lie at upper-arm length ℓ_{SE} from the shoulder, at forearm length ℓ_{EW} from the wrist, and in the half-plane of $\hat{\mathbf{n}}$. Rotating the upper-arm vector $\ell_{SE}\, \hat{\mathbf{e}}_{SW}$ about $\hat{\mathbf{n}}$, where $\hat{\mathbf{e}}_{SW} = \text{unit}(\mathbf{w}^{\text{rb}} - \mathbf{s}^{\text{rb}})$, sweeps the circle of candidate elbows at radius ℓ_{SE} about the shoulder; the valid elbow is where this circle meets the sphere of radius ℓ_{EW} about the wrist. This is exactly SP3 with

$$p_1 = \ell_{SE}\, \hat{\mathbf{e}}_{SW}, \quad k = \hat{\mathbf{n}}, \quad p_2 = \mathbf{w}^{\text{rb}} - \mathbf{s}^{\text{rb}}, \quad d = \ell_{EW},$$

for which $R(k,\theta)\,p_1 - p_2 = \mathbf{e}^{\text{rb}} - \mathbf{w}^{\text{rb}}$, so the SP3 objective $\|R(k,\theta)\,p_1 - p_2\| = d$ enforces $\left\|\mathbf{e}^{\text{rb}} - \mathbf{w}^{\text{rb}}\right\| = \ell_{EW}$ exactly. Of the at most two roots we keep $\theta_{SEW} > 0$, the branch matching the human elbow half-plane, and recover $\mathbf{e}^{\text{rb}}$ via Eq. (3).

The downstream solvers WARP invokes are themselves subproblem-based— SEW-Mimic [16] recovers the arm joints from the corrected skeleton via SP1 and SP2, and ik-geo [10] solves the torso pose—but the cross-embodiment alignment introduces only the single SP3 call above.

G. Lazy Mobile-Base Filter

The goal of lazy base tracking is to separate *what* the base should do from *how* small upper-body motion is absorbed. In the main text, the feature is that the torso handles fine adjustment while the base moves only for genuine relocation; here we specify the filter that implements that behavior. Let $\mathbf{q}_d = (\mathbf{p}_d, \theta_d) \in SE(2)$ be the base pose implied by the human upper-body target and $\mathbf{q}_b = (\mathbf{p}_b, \theta_b)$ the filtered base target maintained by WARP. With deadband radii $\delta_{xy}, \delta_\theta$, define the deadbanded error

$$\tilde{\mathbf{e}}_{xy} = \max\big(0,\, \|\mathbf{p}_d - \mathbf{p}_b\| - \delta_{xy}\big)\,\hat{\mathbf{e}}_{xy}, \quad \tilde{e}_\theta = \operatorname{sign}(e_\theta)\,\max\big(0,\, |e_\theta| - \delta_\theta\big), \tag{6}$$

where $\hat{\mathbf{e}}_{xy}$ is the unit error direction and e_θ the wrapped yaw error. The filter then integrates

$$\ddot{\mathbf{p}}_b = \omega_n^2\,\tilde{\mathbf{e}}_{xy} - 2\zeta\omega_n\,\dot{\mathbf{p}}_b, \qquad \ddot{\theta}_b = \omega_n^2\,\tilde{e}_\theta - 2\zeta\omega_n\,\dot{\theta}_b, \tag{7}$$

with natural angular frequency $\omega_n = 2\pi f_n$ and damping ratio ζ. Inside the deadband the spring force vanishes and the base coasts to rest under damping alone, so jitter and small sway never reach the wheels; outside it, the spring smoothly draws the base toward the target. We use $\zeta = 1$ for non-overshooting tracking, $f_n \approx 1.5\,\text{Hz}$, and deadband radii $\delta_{xy} = 0.05\,\text{m}$ and $\delta_\theta = 0.1\,\text{rad}$.

H. Real Robot System

We deploy the RB-Y1 humanoid (Fig. 3), whose kinematics mirror the human upper body: a holonomic mobile base, a 6-DoF torso, and two 7-DoF arms. The torso lets the upper body reposition without moving the base, improving manipulation stability and extending the reachable workspace beyond an arm-only configuration. Each arm terminates in a 12-DoF five-fingered XHand, of which an internal pose retargeter was used to convert hand poses to robot finger motions. We command the robot through joint-impedance control at $100\,\text{Hz}$ over a wired Ethernet connection. Visual observations come from a head-mounted Project Aria Gen 2 glasses streaming at $10\,\text{Hz}$. Because this work does not address the visual embodiment gap, we track each manipulated object with an AprilTag instead of raw pixels, using the Quest cameras during data collection and the Aria cameras during rollout, and express its pose in the robot upper-body frame (Sec. C). Open-loop replay has no corrective feedback and is therefore sensitive to state-estimation error, so to decouple execution from wheel-odometry drift we measure robot pose with a Vicon motion-capture system. This provides the precise localization that replayability (Sec. III) requires.

I. Policy

We instantiate our policy using HPT [24], a transformer-based action-chunk model that represents observations and actions as tokens and predicts the flow-matching velocity field [18] over future actions. We use HPT mainly as a standard policy backbone, and introduce a body-hierarchy bias that mirrors the structure of our retargeting pipeline.

Specifically, we decompose each action chunk into base, torso, arm, and hand components,

$$\mathbf{a}_{1:H} = [\mathbf{a}^b, \mathbf{a}^\tau, \mathbf{a}^r, \mathbf{a}^h]_{1:H}.$$

Rather than treating all body blocks as fully exchangeable, we impose a proximal-to-distal ordering

$$b \preceq \tau \preceq r \preceq h,$$

where proximal motion provides coarse support and reachability, while distal motion refines interaction. The noisy action chunk is tokenized along both time and body block, and we apply a block-causal attention mask

$$M_{ij} = \begin{cases} 0, & \text{if } j \preceq i, \\ -\infty, & \text{otherwise.} \end{cases}$$

No causal mask is applied along the temporal horizon, allowing the model to reason over the full action chunk jointly. This preserves the standard HPT flow-matching update and inference cost, while encouraging proximal-to-distal information flow in whole-body action prediction. In implementation, low-dimensional observations and hand states are first encoded with MLP encoders and then fused with the action tokens. We train the policy with a batch size of 32 and evaluate the checkpoint from epoch 599. The final model remains lightweight, with approximately 11M trainable parameters.

J. Evaluation Metrics

a) Metrics Definition: We report 18 metrics covering palm tracking, body-pose tracking, joint-limit feasibility, collision, motion deviation, solver variation, and solve time. All position errors are reported in millimetres, angular errors in degrees, and solve time in milliseconds per frame. Unless stated otherwise, each metric is first computed per demonstration over valid post-warmup frames, then averaged across demonstrations using a NaN-safe macro mean, so each demonstration has equal weight regardless of length.

Notation. Let $\mathcal{V}$ be the valid frames of a demonstration and $\langle\cdot\rangle$ the NaN-safe macro mean across demonstrations. For arm $s \in \{\text{L}, \text{R}\}$, $e_t = \frac{1}{2}\sum_s \|\boldsymbol{p}^r_{s,t} - \boldsymbol{p}^h_{s,t}\|$ is the per-frame palm error; Q_p denotes the pth percentile; $\angle(R_a, R_b) = \arccos\big(\frac{1}{2}(\operatorname{tr}(R_a^\top R_b) - 1)\big)$ the geodesic angle between two rotations; $\operatorname{wrap}(\cdot)$ maps an angle to $[-\pi, \pi]$; and $\boldsymbol{q} \in \mathbb{R}^{26}$ the robot joint vector with body-group slices $\boldsymbol{q}^g$.

Palm tracking. For each frame and each arm, we compute the Euclidean distance between the robot palm and the corresponding human palm in the demo-world frame, e_t.

Raw palm error. Mean palm position error over all valid post-warmup frames, $\langle \operatorname{mean}_{t\in\mathcal{V}} e_t \rangle$. This measures average palm tracking accuracy but includes catastrophic outlier frames.

Clean palm error. Mean palm error after removing frames whose error exceeds τ, $\langle \operatorname{mean}_{t\in\mathcal{V},\, e_t\le\tau} e_t \rangle$. This reflects typical-frame tracking quality; a large Raw–Clean gap indicates that a small number of failed frames dominate the raw average.

P95 palm error. The 95th percentile of palm error over valid post-warmup frames, $\langle Q_{95}(\{e_t\}_{t\in\mathcal{V}}) \rangle$. This captures the upper tail of the palm tracking error.

P99 palm error. The 99th percentile, $\langle Q_{99}(\{e_t\}_{t\in\mathcal{V}}) \rangle$. This highlights rare but severe tracking failures.

Arm and torso tracking. These metrics evaluate whether the retargeted motion preserves the human elbow and torso configuration, rather than only matching the palm pose.

Elbow error. Mean absolute error between the robot and human SEW elbow swivel angles ψ, $\langle \operatorname{mean}_{t\in\mathcal{V}} |\mathrm{wrap}(\psi_t^r - \psi_t^h)| \rangle$. Lower values indicate better preservation of the human elbow configuration around the shoulder–wrist axis.

Torso error. Mean rotation-angle error between the robot torso and the human upper-body orientation, $e_t^{\mathrm{to}} = \angle(R_t^r, R_t^h)$, giving $\langle \operatorname{mean}_{t\in\mathcal{V}} e_t^{\mathrm{to}} \rangle$. Lower values indicate closer torso pose matching.

Out@15. Fraction of valid post-warmup frames whose torso error exceeds $15°$, $\langle \operatorname{mean}_{t\in\mathcal{V}} \mathbb{1}[e_t^{\mathrm{to}} > 15°] \rangle$. This measures how often the retargeted motion enters a visibly poor torso-pose state.

Joint-limit feasibility. For each joint j with mechanical limits $[\ell_j, u_j]$, the per-frame margin is $m_{j,t} = \min(q_{j,t} - \ell_j,\, u_j - q_{j,t})$. Wheels and joints without known mechanical bounds are excluded.

Limit fraction. Fraction of joint–frame pairs within $10°$ of a mechanical limit, $\langle \operatorname{mean}_{j,t} \mathbb{1}[m_{j,t} < 10°] \rangle$. Lower values indicate fewer near-limit configurations.

Minimum margin. Minimum margin over all joints and valid post-warmup frames, macro-averaged across demonstrations, $\langle \min_{j,\, t\in\mathcal{V}} m_{j,t} \rangle$. Larger is better; a negative value means the planned trajectory exceeds the robot's mechanical limits and would be clipped on hardware.

Self-collision. This metric evaluates physical feasibility by replaying the retargeted robot trajectory in simulation.

Collision fraction. Fraction of valid post-warmup frames in self-collision, $\langle \operatorname{mean}_{t\in\mathcal{V}} c_t \rangle$ with $c_t \in \{0, 1\}$. We check the arms, torso, head, and wheels, while keeping the fingers in a neutral pose. Lower values indicate safer trajectories for hardware deployment.

Normalized motion deviation. We use Nearest-Neighbor Action Disagreement (NNAD) to measure trajectory-stack determinism. For each human-state observation i, let $\mathcal{N}_k(i)$ be its k nearest human-state neighbours drawn from other demonstrations; the group-g disagreement is

$$\mathrm{NNAD}_g \;=\; \Big\langle \operatorname*{mean}_i \; \tfrac{1}{k} \sum_{j\in\mathcal{N}_k(i)} \big\| \boldsymbol{q}_i^g - \boldsymbol{q}_j^g \big\| \Big\rangle. \tag{8}$$

Low NNAD means that similar human inputs map to similar robot outputs, which is important for stable downstream policy learning.

R-arm NNAD. NNAD_g for the right-arm joints. Lower values indicate more deterministic right-arm retargeting.

L-arm NNAD. NNAD_g for the left-arm joints. Lower values indicate more deterministic left-arm retargeting.

Torso NNAD. NNAD_g for the torso joints. Lower values indicate more consistent torso retargeting.

Head NNAD. NNAD_g for the head joints. Lower values indicate more consistent head motion.

Solver variation. These metrics come from a controlled pose-perturbation study. For each sampled input pose, the IK solver is run N times with Gaussian-jittered warm starts, producing outputs $\{\boldsymbol{q}^{(n)}\}_{n=1}^N$; $\langle\cdot\rangle_p$ denotes the mean over sampled poses. Unlike the previous metrics, these isolate solver-only non-determinism.

RMS variation. Root-mean-square per-joint standard deviation across repeated trials (degrees), $\big\langle \sqrt{\frac{1}{J}\sum_j \mathrm{Var}_n\, q_j^{(n)}} \big\rangle_p$. This measures typical per-joint solver uncertainty.

Pair-L2 variation. Average pairwise ℓ_2 distance between repeated outputs, $\big\langle \operatorname{mean}_{a<b} \|\boldsymbol{q}^{(a)} - \boldsymbol{q}^{(b)}\| \big\rangle_p$. This measures the typical disagreement between two solver outputs for the same input pose.

PCA variation. Largest eigenvalue of the output covariance $\Sigma = \mathrm{Cov}_n(\boldsymbol{q}^{(n)})$, $\big\langle \lambda_{\max}(\Sigma) \big\rangle_p$. This measures how strongly solver variation concentrates along a dominant null-space direction.

Solve time. This metric measures solver-only computational cost.

Solve time. Mean time spent inside the solver per frame, excluding file I/O, simulation bookkeeping, and video writing. Lower values indicate faster retargeting.

All metrics are better when lower, except *Minimum margin*, where higher is better.

K. Experiment Setup

1) Real World Environment Setup: All data is collected at 20 Hz from Quest. During the rollout, to accommodate the controller limited performance, we lower the rollout/control freq to 6 Hz (30% of the original speed). This setup is applied to both MINK and WARP.

a) Human data collection: We collect demonstrations with a single Meta Quest headset, with no external motion-capture rig, body-worn markers, or robot in the loop. Unlike traditional teleoperation and UMI-style interfaces, which compress a demonstration to the end-effector's spatial pose,

we capture the operator's whole-body motion: the headset's inside-out tracking returns, per frame, an upper-body kinematic tree together with both hands, each bone a 6-DoF pose in a fixed world frame, plus the root pose, which captures in-place locomotion. The same headset tracks an AprilTag on the manipulated object in its camera frame, and the simultaneous skeleton estimate lets us re-express the object in the operator's upper-body frame. The operator views the scene through passthrough and toggles recording from the in-headset interface (Fig. 1); we log this full-body pose, hand articulation included, at 60 Hz. The resulting raw human dataset is

$$\mathcal{D}_{\text{raw}} = \left\{ \left(\mathbf{T}^{\text{hm}\leftarrow\text{obj}}(t),\, \mathcal{H}(t) \right) \right\}_{t=1}^{N},$$

where $\mathbf{T}^{\text{hm}\leftarrow\text{obj}}(t) \in \text{SE}(3)$ is the object pose in the operator's upper-body frame and $\mathcal{H}(t) \in \mathbb{R}^{d_{\mathcal{H}}}$ is the human motion feature (hand and finger poses).

Expressing the object in the upper-body frame is what lets each human observation transfer to the robot without a world frame. Because our cross-embodiment alignment places a robot at the anchor with its end-effector coinciding with the operator's, the fixed anchor transform yields the object pose in the robot's equivalent upper-body frame, $\mathbf{T}^{\text{rb}\leftarrow\text{obj}}(t)$ — exactly what the robot would observe while reproducing the motion. We retarget each frame to the robot embodiment with WARP, mapping the human motion feature to a matched robot proprioceptive state $\tilde{s}_t \in \mathbb{R}^{d_s}$ and a control action $u_t \in \mathbb{R}^{d_a}$. Combined with the frame-transformed object pose $\mathbf{T}^{h_c\leftarrow o}(t) \in \text{SE}(3)$ from Sec. III, this yields the robot training dataset

$$\mathcal{D}_{\text{robot}} = \left\{ \left(\left(\mathbf{T}^{h_c\leftarrow o}(t),\, \tilde{s}_t \right),\, u_t \right) \right\}_{t=1}^{N}.$$

At deployment, the robot computes $\mathbf{T}^{h_c\leftarrow o}(t)$ from its own onboard perception and reads the proprioceptive feature s_t from its own state, then queries the policy for the next action

$$a_t = \pi_\theta(o_t), \qquad o_t = \left(\mathbf{T}^{\text{hm}\leftarrow\text{obj}}(t),\, s_t \right).$$

Because s_t and $\tilde{s}_t$ share the same space, and $\mathbf{T}^{\text{hm}\leftarrow\text{obj}}(t)$ is computed identically at training and deployment, the policy trained on human data transfers to the robot without fine-tuning.

b) Replay Experiment Setup: We retarget offline human motion into 20 Hz robot trajectories. Both open-loop replay and policy rollout run at a 6 Hz command rate—30% of the recorded speed—which keeps the joint-impedance controller's steady-state tracking error negligible; replay executes the full 20 Hz trajectory at this rate, and policies are trained on the same 20 Hz data.

A motion-capture system provides accurate, drift-free localization of the robot base. Open-loop replay carries no perception feedback to correct an initial mismatch, so each trial must begin from the same robot–object relative configuration the retargeting assumes, including the adaptive offset WARP applies across embodiments. We establish this configuration once and reproduce it across trials by fixing the robot start pose and object placement in the mocap frame with ground markers.

2) Real-world Per-Task Scoring Criteria: Each manipulation task is scored on a graded success scale; the replay-only fridge task is scored as a binary completion rate.

Pick-up-laundry. We score grasp and release symmetrically: 1.0 when both hands complete the phase, 0.5 for one, 0 otherwise.

Push-cart. We score on a graded scale: 1.0 for a clean two-handed push, 0.5 for single-handed, 0.25 for a push with incidental body contact, and 0 for missing or hitting the cart without moving it.

Rotate-box. Progress is discretized by rotation angle: $60°-90° = 1.0$, $30°-60° = 0.5$, $0°-30° = 0$.

L. DexMimicgen Experiment setup

We evaluate WARP and MINK retargeting on the same GR1 DexMimicGen demonstrations and convert them into RB-Y1 policy-training data. The source demonstrations are collected from three bimanual tasks: `two_arm_coffee`, `two_arm_pouring`, and `two_arm_can_sort_random`. Each demonstration contains the recorded GR1 MuJoCo state, action commands, and environment metadata. We first retarget each GR1 demonstration into a body-centric representation containing shoulder, elbow, wrist, torso, head, and hand information. For SEW, the extracted shoulder–elbow–wrist geometry is later solved online by the RB-Y1 SEW solver during playback. For MINK, the corresponding RB-Y1 joint targets are precomputed and stored, so data collection does not require online IK solving.

During collection, each retargeted trajectory is replayed in a dual-robot robosuite environment. The RB-Y1 robot executes the retargeted motion, while the original GR1 motion is optionally overlaid as a ghost robot for visualization. The robot base is fixed using a canonical base placement computed from a reference frame, which avoids per-demonstration base drift and keeps the comparison between WARP and MINK consistent. Finger retargeting is bypassed: the original 6-DoF Fourier hand command from GR1 is copied directly and deterministically expanded by the Fourier hand controller into the actuator command used by RB-Y1.

We keep only successful demonstrations for policy training. A trajectory is considered successful only if the task succeeds during execution and remains successful at the final frame. After collection, we further intersect the WARP and MINK datasets using the original demonstration index, so both methods are trained and evaluated on matched demonstrations. Finally, we slice a balanced subset of the first $N = 50$ demonstrations for each task and method to construct the final policy-training datasets.

1) BONES-SEED Dataset Retargeting Setup: BONES-SEED [4] is a large-scale skeletal human motion dataset for everyday humanoid behaviors, released in three motion-format subsets. *SOMA Uniform* standardizes all motions to a single skeleton for consistent topology and scale, supporting large-scale processing and retrieval. *SOMA Proportional* preserves

per-actor body proportions through shape parameters, capturing morphology-dependent motion geometry. *Unitree G1* retargets trajectories to the G1 humanoid as MuJoCo-compatible joint sequences for simulation-based imitation learning. We use SOMA Proportional: its actor-specific morphology preserves real human motion geometry more faithfully than the uniform representation, the right fit for studying human-to-robot retargeting.

a) Dataset sampling.: We evaluate on a curated subset of the SOMA Proportional motions, drawn from the full BONES-SEED metadata by three filters. A *taxonomy* filter selects motions by the activity labels provided in the original BONES-SEED metadata: we keep the `Everyday` package (5,816 motions), which the dataset subdivides into the five categories `Household`, `Unusual Locomotion`, `Consuming`, `Object Interaction`, and `Environments`. A *description* filter then restricts to manipulation-relevant motions: we concatenate each motion's four natural-language description fields (the `content_natural_desc` columns; the short and technical descriptions are not searched), lowercase the text, and keep the motion if it contains any of the substrings `table`, `everyday`, or `object` and none of `sit` or `seated`. Matching is case-insensitive substring containment rather than whole-word, so `table` also matches *tabletop* and *vegetables* while `sit` also removes *transit* or *position*; this loose rule is intentional and mostly captures additional table-adjacent motions. Finally, a *filename* filter drops high-reaching motions—those whose filename contains `_high_`, such as reaching into high cupboards or fridges—since these often exceed RB-Y1's vertical workspace and would measure reach-envelope violations rather than retargeting quality. We apply no mirror filtering: original and mirrored sequences both pass whenever they satisfy the predicates, preserving paired left- and right-handed variants. The final subset contains 514 motions—257 original and 257 mirrored—comprising 135,896 retargeted robot frames at the IK solver output rate.

M. Additional Results

1) BONES-SEED subset Results across all metrics: Table III shows complete metrics results on the 514 trajectories subset.

2) Random Sample BONES-SEED Dataset Results: Table IV shows complete metrics results on the a 7.6k trajectories more completed subset.

a) Larger-scale evaluation on the 7.6k BONES-SEED subset.: We evaluate all four variants on the broader two-per-task BONES-SEED subset: 7,623 motions spanning the full BONES-SEED activity taxonomy—everyday manipulation, locomotion, idle motions, jumps, and unusual motions. This is well beyond the 514-motion subset used in the main paper. The main-paper trends hold, but the gaps widen sharply (Table IV).

N. Limitations

a) Kinematic assumptions.: Although WARP enables accurate and consistent retargeting from offline human demonstrations, our current formulation remains primarily kinematic. The retargeter generates robot joint trajectories that satisfy geometric constraints, including palm alignment, SEW consistency, and whole-body pose similarity, but it does not explicitly model controller error or robot dynamics during execution. In our analysis, we assume that the low-level controller can accurately track the retargeted reference motion. This assumption helps isolate the retargeting problem, but it abstracts away actuator bandwidth limits, latency, contact dynamics, and disturbances caused by object interaction. Therefore, the reported retargeting accuracy should be interpreted as the quality of the generated reference trajectory rather than a complete guarantee of real-world execution accuracy. In practice, even a geometrically valid trajectory can fail if the controller lags, if contact moves the object unexpectedly, or if the base and torso cannot realize the desired motion smoothly. Future work should incorporate controller-in-the-loop evaluation and dynamics-aware retargeting, potentially with feedback or residual controllers that compensate for tracking errors during contact-rich manipulation.

b) Simplified policy and perception.: Our current policy and perception setup is also simplified. To focus on retargeting and action learning, we use low-dimensional object-state inputs based on AprilTag tracking rather than learning directly from raw visual observations. This controlled setup reduces perception noise and makes the real-world experiments more reproducible, but it limits the system's deployment scope. In realistic home environments, objects may not carry fiducial markers, tags may be occluded during manipulation, and object appearance can vary across scenes. As a result, the current policy does not fully address marker-free visual generalization. A natural next step is to replace AprilTag-based tracking with learned object pose estimation, dense visual features, or RGB-D perception. External side-view cameras could also provide complementary scene-level observations, especially when the robot's egocentric view is occluded by its hands, arms, or manipulated objects.

c) Human data quality.: Finally, the human data collection device introduces its own noise. We use a Quest headset because it provides an accessible and scalable way to collect whole-body motion without a full motion-capture system. However, we observe that some motion error and jerkiness in the retargeted trajectories can originate from the collection device itself. Inside-out tracking may introduce jitter, drift, discontinuities, or inaccurate limb estimates, especially during fast motion, occlusion, or unusual body poses. Since our pipeline is offline, these artifacts can propagate directly into the robot trajectory and then into the policy supervision. Future work could improve data quality by fusing multiple sensing sources. For example, an external side-view camera or multi-view RGB-D setup could provide a more stable estimate of the human skeleton, while a lighter and more integrated wearable device, such as Aria glasses, could reduce the burden of wearing a VR headset and better preserve natural human motion.

		Palm tracking ↓				Arm / torso tracking ↓			Joint-limit / feasibility	
Method	**JL**	**Raw** mm	**Clean** mm	**P95** mm	**P99** mm	**Elbow** deg	**Torso** deg	**Out@15** frac.	**Limit** frac.	**Margin** ↑ deg
SEW-M	off	178.979	–	201.056	203.800	2.669	**1.189e-6**	**0.000**	0.0126	7.108
MINK-EF	off	0.701	0.609	1.853	3.389	79.110	20.680	0.6247	0.1610	-54.95
MINK-TE	off	18.557	1.360	73.980	115.400	8.966	3.857	0.0266	0.0852	-70.11
WARP	off	**0.0046**	**0.0005**	**0.0465**	**0.0621**	**0.0105**	1.220e-6	**0.000**	**0.0047**	**13.02**
SEW-M	on	215.641	–	272.842	285.700	5.775	5.329	0.1624	0.0131	4.804
MINK-EF	on	0.751	0.520	2.478	4.298	39.250	19.430	0.6113	0.1247	-1.000
MINK-TE	on	19.492	1.352	69.345	105.300	6.881	5.128	0.0491	0.0420	1.648
WARP	on	24.048	0.154	82.036	100.600	3.344	4.451	0.1298	0.0060	9.559

		Collision ↓	Normalized motion deviation ↓			Solver variation ↓			Solve time ↓
Method	**JL**	**Coll.** frac.	**R-arm** –	**L-arm** –	**Torso** –	**RMS** deg	**Pair-L2** –	**PCA** eig.	**Solve** ms
SEW-M	off	0.243	0.6578	0.6713	0.4879	6.828e-14	**0**	1.198e-25	14.83
MINK-EF	off	0.977	0.6951	0.7052	0.3677	3.117	19.43	173.5	192.1
MINK-TE	off	0.640	1.603	1.776	1.026	6.106	34.33	1119.0	235.8
WARP	off	0.163	0.6825	0.7103	0.2888	6.657e-14	**0**	1.140e-25	6.781
SEW-M	on	0.084	**0.6527**	**0.6567**	0.4419	6.561e-14	**0**	1.089e-25	**4.978**
MINK-EF	on	0.222	0.7233	0.7321	0.3543	3.169	19.61	227.3	192.5
MINK-TE	on	0.478	0.9570	0.9569	0.4539	4.964	28.38	613.5	234.4
WARP	on	**0.017**	0.7270	0.7434	**0.2659**	**6.456e-14**	**0**	**1.062e-25**	5.430

TABLE III: Quantitative simulation retargeting results for the highlighted variants. The table is split into two rows for readability: the top row reports tracking and joint-limit feasibility metrics, while the bottom row reports collision, normalized motion deviation, solver variation, and solve time. For all metrics, lower is better except **Margin**, where larger is better. Best results are shown in **bold**; second-best results are underlined.

		Palm tracking ↓				Arm / torso tracking ↓			Joint-limit / feasibility	
Method	**JL**	**Raw** mm	**Clean** mm	**P95** mm	**P99** mm	**Elbow** deg	**Torso** deg	**Out@15** frac.	**Limit** frac.	**Margin** ↑ deg
SEW-M	off	200.2	–	252.0	317.8	2.672	1.060	0.01311	0.02643	**-4.891**
MINK-EF	off	248.0	**0.7856**	432.0	450.4	83.43	39.46	0.7340	0.1603	-61.74
MINK-TE	off	310.3	1.586	654.4	797.3	33.25	17.85	0.2221	0.1890	-270.0
WARP	off	**11.91**	1.51	**43.77**	**78.98**	**1.144**	**0.6469**	**0.00706**	**0.02343**	-7.736

		Collision ↓	Normalized motion deviation ↓			Solver variation ↓			Solve time ↓
Method	**JL**	**Coll.** frac.	**R-arm** –	**L-arm** –	**Torso** –	**RMS** deg	**Pair-L2** –	**PCA** eig.	**Solve** ms
SEW-M	off	**0.1028**	**1.061**	**1.014**	1.125	6.828e-14	**0**	1.198e-25	**7.626**
MINK-EF	off	0.9203	1.266	1.266	**0.7478**	3.117	19.43	173.5	382.7
MINK-TE	off	0.4639	5.089	5.128	3.370	6.106	34.33	1119.0	452.1
WARP	off	0.1112	1.298	1.234	0.7921	**6.657e-14**	**0**	**1.140e-25**	10.52

TABLE IV: Quantitative retargeting results on the 7,623-motion BONES-SEED two-per-task subset. The table is split into two rows for readability: the top row reports tracking and joint-limit feasibility metrics, while the bottom row reports collision, normalized motion deviation, solver variation, and solve time. For all metrics, lower is better except **Margin**, where larger is better. Best results are shown in **bold**; second-best results are underlined.