# Cross-Embodiment Transfer of a Frozen Humanoid Whole-Body Controller via Analytic Codec and LoRA Adapters

Sitarama Chekuri  Claude Fable 5 (Anthropic)

Draft v0.1 — 2026-08-16

**Abstract**

Whole-body tracking models have become foundation assets for humanoid robots, yet each new platform still tends to receive its own controller, trained from scratch at a cost few robotics labs can spend. Recent work showed that a pretrained tracker can be adapted to a new humanoid for a small fraction of the original training cost. We ask a stronger question: how far can a new robot get if the pretrained platform is never trained at all? We transfer a frozen, publicly released whole-body controller (GEAR-SONIC, trained on the Unitree G1) to a different humanoid (AgiBot X2 Ultra) using only a closed-form joint-space codec and low-rank adapters on a single decoder, leaving every encoder, the quantized motion representation, and the platform's motion prior untouched. Because the two skeletons are closely matched by design, the codec alone already yields a surviving zero-shot controller; the adapters, trained overnight on one 8-GPU node — roughly two percent of the platform's cited training compute — then absorb the dynamic gap. The result is not parity but reversal: on out-of-distribution content the transferred controller outperforms a tracker natively trained for the target robot, while standard in-distribution benchmarks cannot tell the two apart. The same recipe applied to the platform's own body specializes it to a bank of demonstration motions without measurable forgetting. Freezing the platform also makes its limits legible, and we characterize both: an embodiment cost floor that no adapter training buys back, and a causally verified information bottleneck — signals the platform never learned to encode cannot be recovered by any adaptation downstream of the freeze. We release the models, the codec, and the evaluation protocol. Video comparisons and models: https://sonic-agibot-x2.github.io/sonic-transfer/

## 1 Introduction

A humanoid robot is no longer limited by what its own team can afford to teach it. Foundation-scale whole-body controllers now exist — trained on hundreds of hours of motion capture, at compute budgets measured in thousands of GPU-hours — and some are publicly released. Yet a released controller remains welded to the body it was trained on: its observations, actions, and learned dynamics all speak one robot's language. For a team bringing up a different robot under real budget constraints, the asset is tantalizing but unusable as-is — retraining it at its native scale is out of reach, and the standard alternative remains training a robot-specific controller from scratch at whatever scale the budget allows.

The gap between those two scales is the subject of this paper. Our target robot's incumbent controller was trained the way most real-world robot controllers are: a warm-started curriculum, accumulated over weeks of experiments, totaling on the order of 1,500 GPU-hours. The platform we borrow was trained once, by its authors, at several times that scale, and we choose to keep it frozen — not because it cannot be fine-tuned (Section 9 fine-tunes it), but because freezing is the

experiment: it isolates what a pretrained motion prior is worth, unmodified, to a body it has never seen. The question is whether that frozen platform, wrapped but never retrained, can serve the new robot better than the specialist trained for it on purpose.

Our answer is affirmative, and by a wide margin on the measurements that matter. We transfer the released GEAR-SONIC controller (Luo et al. 2026) to the AgiBot X2 Ultra with two components: a *closed-form codec* — a per-joint affine table, fitted once, no network — that translates between the two robots' joint spaces; and *low-rank adapters* on the platform's dynamics decoder, the only trainable parameters in the system, amounting to a quarter of one percent of the platform's weights. Everything else — three motion encoders, the quantized motion representation that constitutes the platform's motion prior, and its kinematic decoder — remains bit-identical to the release.

Three findings organize our results. First, *the transfer wins where it matters and ties where it doesn't*: on out-of-distribution motion content the adapted controller beats the natively trained incumbent by ten success points with a better error tail, while on in-distribution benchmarks the two are statistically indistinguishable — a blindness of in-distribution evaluation that we believe generalizes beyond this paper. Second, *the recipe is symmetric*: with the codec removed, the same adapters specialize the platform on its own native body, saturating a bank of previously failing demonstration motions while slightly improving the platform's original benchmark — specialization without forgetting. Third, *freezing makes limits legible*: we characterize an embodiment cost floor — an error offset invariant across every training lineage and budget we tested, attributable to the target's weaker actuators and the retargeting itself — and an information bottleneck, causally verified by a reference-perturbation probe, showing that signals the platform never learned to encode cannot be recovered by any adaptation downstream of the freeze.

Our contributions:

1. A cross-embodiment transfer recipe for a frozen whole-body controller — analytic codec plus decoder-side LoRA — that outperforms a natively trained tracker on out-of-distribution content at roughly two percent of the released platform's training compute.
2. The finding that in-distribution benchmarks are blind to this difference, motivating the three-benchmark evaluation protocol we release.
3. A characterization of last-mile training dynamics under a frozen platform: out-of-distribution success follows a measurable slope, peaks, and declines under continued fine-tuning, yielding a practical gate-and-stop procedure.
4. Two boundary results mapping what adaptation cannot do: an embodiment cost floor invariant across training lineages, and a causally verified information bottleneck for reward-invisible signals.
5. Evidence that the recipe is embodiment-agnostic in both directions: on the platform's own body it saturates a demanding specialization set while slightly improving the original benchmark.

## 2 Related Work

**Scaled motion tracking.** GEAR-SONIC (Luo et al. 2026) established that scaling model capacity, data, and compute yields a generalist humanoid motion tracker with strong generalization to unseen motions, and released a checkpoint, code, and a large portion of its training corpus. We use the released checkpoint as a frozen platform and never modify it. Our companion work transfers the same platform's *kinematic planner* to the same target robot with purely output-side machinery; Section 4 discusses why the whole-body controller demands the opposite placement.

**Cross-embodiment control.** Approaches to sharing control across morphologies include multi-embodiment training with morphology randomization, topology-aware architectures, and unified observation–action spaces. These build generalists from scratch; our setting is the complementary one — a single-embodiment specialist asset, already trained, that must serve a new body.

**Parameter-efficient adaptation.** LoRA (Hu et al. 2021) and related methods freeze most pretrained weights and learn low-rank updates; we adopt LoRA in its most restrictive placement — one decoder — and use zero-initialized adapters so optimization starts from the platform's exact behavior.

### 2.1 Relation to Any2Any

The closest prior work is Any2Any (Yang et al. 2026) (LimX Dynamics), which adapts the same pretrained platform — GEAR-SONIC — to new humanoids, and arrives at a strikingly similar decomposition: a kinematic alignment stage that places the target robot's observations and actions into the source policy's semantic space, followed by LoRA-based dynamics adaptation. We share their premise, their two-stage structure, and their parameter-efficiency result. The two works differ in what they align, what they freeze, and what they claim.

*What is aligned.* Any2Any's benchmark spans four platforms across two morphological families and 19–31 degrees of freedom; bridging that spread requires a learned, two-level alignment module. Our target was designed in the source robot's image — same kinematic tree, joint-for-joint correspondence — and this similarity converts alignment from a learning problem into a closed-form one. The consequence shows up immediately in the zero-shot regime. Any2Any's position is that "even between similar humanoids, structural and dynamic differences make direct deployment unreliable" — and across their morphologically diverse pairs it surely is. Our measurement qualifies that statement at the extreme end of similarity: for a pair matched joint-for-joint by design, codec-only zero-shot transfer not only survives — it tracks in-regime motion with *better* joint fidelity than the target's own natively trained specialist, before a single gradient step (Section 5.1). We are explicit about the attribution: this is a gift of morphological similarity, not of method, and we report it as a precondition measurement.

*What is frozen.* Both works freeze the pretrained backbone — the distinction is the freeze *boundary*. Any2Any inserts adapters into the modules with the largest source–target discrepancy, selected per robot pair and spanning actor and critic, and its alignment stage is itself a learned module. We fix the boundary once, maximally: every encoder, the quantized motion representation, and the kinematic decoder remain bit-identical to the release; exactly one decoder carries adapters; alignment is closed-form with nothing learned in it. The stricter contract is what makes the platform's properties legible: because nothing upstream can move, generalization that survives transfer is provably the platform's, the embodiment floor of Section 7 is attributable to the body rather than to forgetting, and the information bottleneck of Section 8 is measured against a fixed representation.

*What is claimed.* Any2Any's central result is parity at low cost. Ours is complementary and stronger where similarity permits it: on out-of-distribution content, the frozen generalist plus last-mile adapters *outperforms* native training — which, we note, is no strawman baseline: the incumbent was built under real budget constraints as a three-stage warm-started curriculum (2k locomotion clips, then a 30k-motion corpus, then the full 130k corpus; roughly 1,600 GPU-hours cumulatively). The transfer used roughly an eighth of that budget, skipped the curriculum entirely, and generalizes

better.[1]

# 3 Setup: Platform, Bodies, and Benchmarks

**The platform.** GEAR-SONIC is a token-based encoder–decoder controller trained on the Unitree G1 (29 DoF) from 100M+ frames of motion capture. Reference motion enters through one of three encoders (robot-frame motion, VR teleoperation targets, or SMPL human motion), is compressed through a finite-scalar-quantized token bottleneck (Mentzer et al. 2023), and drives two decoder heads — one reconstructing reference trajectories, one emitting joint-space actions (the paper calls the latter the robot-control decoder; we adopt the released code's naming, kinematic and dynamics decoders). The released checkpoint totals ~26M policy parameters. We treat it as immutable.

**The bodies.** The target, AgiBot X2 Ultra, was designed with a skeleton closely matched to the G1: same kinematic tree, joint-for-joint naming correspondence. The match ends at the actuators: the X2's per-kilogram torque capability is roughly half the G1's at the ankles and waist and far less at the wrists. The morphological similarity is a deliberate design precondition of this work and is reported as such; the actuator gap is what adaptation must absorb — and, as Section 7 shows, partially cannot.

**Baselines.** The *incumbent* is the target robot's own tracker, trained natively as described above, and deployed on the physical robot prior to this work. The *released checkpoint* itself, evaluated on the native G1 body, provides the platform ceiling.

**Benchmarks.** Our protocol uses three benchmarks with distinct roles, all evaluated under strict tracking-success gates with identical termination criteria, plus a second-referee survival check in a different physics engine for any checkpoint considered for deployment. *novel500* (in-distribution): 500 category-stratified clips from the corpus family, held out from all our training. *hard300v3* (tail): 300 clips curated for difficulty. *PHUMA* (out-of-distribution): a 1,931-clip stratified sample of the external PHUMA dataset (Lee 2025), which no model in this paper — including the platform — was trained on; retargeted to each body.

## 3.1 Calibrating our benchmark against the platform's own

A transfer paper faces a quiet credibility problem: its benchmarks are its own. We bridge our scale to the platform authors' with the one instrument both sides share — the released model itself. On their held-out split they report 98.7% success and 23.2 mm mean joint error; the identical model scores 98.2% and 25.7 mm on our novel500. Within half a success point and 2.5 mm, the released model cannot tell our benchmark apart from its authors' — certifying novel500 as difficulty-comparable and licensing discussion in the platform's own units. Two honesty notes: the corpus family overlaps the platform's training data, so its score here is a favorable-condition ceiling — acceptable because any bias flatters the baseline we aim to beat, not us; and the platform's authors label their split by content novelty, while in our protocol novel500 plays the in-distribution role for the target-robot models under test.

[1] Cited figures for SONIC's training compute range from ~9,000 GPU-hours (Any2Any's citation for the release) to 21,000 for the full program; we adopt the conservative lower figure as our denominator throughout.

### 3.2 A reproduction we could not complete

One calibration attempt did not close, and we report it as such. The platform's authors also evaluate on PHUMA and report 97.0% success; running the released checkpoint on our 1,931-clip sample under our protocol, we measure 82.8%. We do not treat this as a contradiction of their result — we were unable to reproduce it under our setup, and we may well have missed something. The candidate explanations are ordinary: their PHUMA evaluation runs in MuJoCo ("all methods are evaluated in MuJoCo," their protocol note), ours in the training simulator; our sample is deliberately tail-heavy and category-stratified; termination conventions and retargeting may differ; and they note PHUMA derives from a different retargeting pipeline than their corpus. Any of these, or an error of ours we have not found, could account for the gap. The consequence is methodological: every PHUMA comparison we draw is internal — all models scored on the identical slice, same simulator, same referee, same machine — and the authors' PHUMA figure appears only in this note, never in our tables.

## 4 Method: Transferring a Frozen Whole-Body Controller

We treat the entire released system as a fixed platform. Nothing in it is retrained: not the encoders, not the quantizer, not the kinematic decoder, and not the backbone of the dynamics decoder. The transfer problem is thereby reduced to two questions: how to present a new robot's reference motion and proprioception to a model that has only ever seen another body, and how to reshape the model's actions into commands the new body can execute.

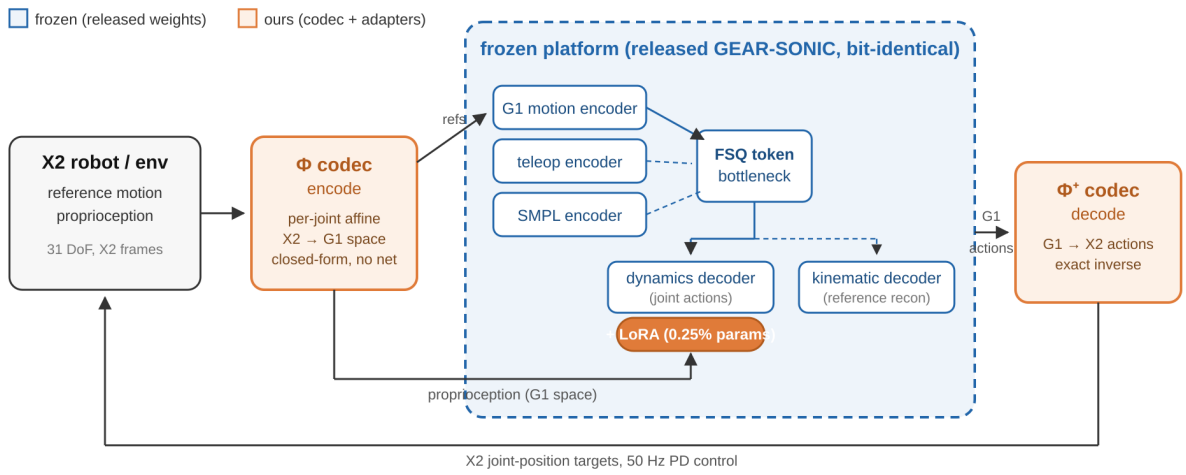


Figure 1: System architecture: the frozen platform (three encoders, FSQ token bottleneck, two decoders) untouched; the analytic codec wrapping its G1-frame interface on both sides; LoRA adapters riding the dynamics decoder.

### 4.1 An analytic codec between bodies

The first question is answered without learning. The skeleton match makes an *analytic codec* sufficient: a per-joint affine map, fitted once from a paired retargeting corpus, that translates X2 joint space into the G1 joint space the platform expects (the encode direction) and translates the platform's G1-frame actions back into X2 commands (the decode direction). The codec also

carries the small set of conventions that differ between the bodies — a wrist axis-naming swap, a stride-length scale, and a height affine — and, being bijective on the shared subspace, encode and decode remain exact inverses. It is exported alongside the model as a calibration sidecar: a table of per-joint coefficients, not a network. Everything the platform sees is G1; everything the robot receives is X2.

A codec alone, however, treats the two bodies as if they differed only in geometry. They do not. The X2's actuators are substantially weaker at the ankles, waist, and wrists, and its dynamics under the same commands differ accordingly. This dynamic gap is what the learned component must absorb.

### 4.2 Last-mile adaptation with decoder-side LoRA

We attach low-rank adapters to the linear layers of the dynamics decoder only — the last stage between the frozen motion prior and the robot's joints. The adapters are initialized to exact zero effect, so optimization starts from the platform's own behavior, and they are the only trainable parameters in the system: a quarter of one percent of the platform's parameter count. Training is standard on-policy reinforcement learning in simulation with the platform's original tracking objective; the reference motions are the target robot's own retargeted corpus, passed through the codec. Intuitively, the frozen prior proposes; the adapter disposes — it learns how the proposal must be bent to survive the new body's dynamics, and nothing more.

**Why adapters inside the model, when the planner transfer kept them outside.** In the companion work introduced in Section 2 — transferring this platform's kinematic planner to the same robot — the learned component sits entirely outside the frozen model: small residual heads post-process the planner's outputs. That placement was correct there for a reason that fails here. A planner emits reference trajectories — kinematic quantities whose small errors are absorbed by the downstream tracker, which supplies the stability margin; output-space correction is therefore both sufficient and safe. A whole-body controller enjoys no such downstream safety net. Its outputs are actions in closed loop with the body's dynamics at control rate, and the transfer mismatch is dynamic: the correction a weaker ankle needs depends on posture, contact state, and velocity — context that lives in the decoder's hidden features, not in the action vector an external module would see. An output-side corrector would have to invert the plant mismatch open-loop and blind. The companion work's own central finding seals the argument: offline fit and closed-loop viability diverge — externally fitted correctors that matched targets best destabilized the closed loop. Dynamics-side adaptation must therefore be trained *in* the closed loop and must live *inside* the decoder where the state context is; low-rank adapters are the least invasive way to be inside — zero-initialized, reversible by deletion, and leaving every frozen weight bit-identical.

Two properties of this arrangement matter for everything that follows. First, because the prior is frozen, whatever generalization the platform acquired at scale is preserved by construction — the adapter cannot overwrite it, only modulate its expression. Second, because the adapter sits downstream of the token bottleneck, it can only work with information the platform chose to encode. Both properties are double-edged, and Sections 7 and 8 measure their costs.

### 4.3 Training schedule and gated evaluation

Adaptation proceeds in two phases. A *breadth* phase trains the adapters on the full retargeted corpus under the platform's adaptive motion sampler, and is stopped when in-distribution improvement plateaus. A *polish* phase then rebalances sampling toward uniform corpus coverage with a small, fixed share of each batch reserved for a curated bank of target behaviors. The polish phase pursues

two objectives at once: to keep improving the curated behaviors after overall progress has plateaued, and — for motions that may never fully succeed on this body — to make the ones that do survive track better across the board. Throughout, we checkpoint frequently and evaluate every checkpoint under two independent referees: the training simulator with strict tracking-success gates, and a second physics engine with a survival criterion mirroring the deployment stack. A checkpoint is only eligible for deployment if both accept it. Out-of-distribution success is tracked at every gate, and the run is stopped by rule when it declines at two consecutive gates; Section 6 shows this rule is not a formality.

### 4.4 Selecting the fine-tune motions

The polish bank's origin is candidly pragmatic: demonstrations were scheduled before any training run could be carried to full convergence, so we curated the set of behaviors the demonstrations actually needed — basic locomotion at demonstration speeds, dance, and boxing — and asked the polish phase to make those good enough, soon enough. The bank is deduplicated against the corpus and injected with override semantics — where a bank motion shares an identity with a corpus clip, the bank's version replaces it. The injection share is deliberately small and flat: a fixed fraction of each batch, exempt from the adaptive sampler's failure-chasing, so the bank acts as steady rehearsal rather than a curriculum takeover. Two selection principles emerged from failed alternatives. First, feasibility screening: motions whose references a flat, prop-free scene cannot physically satisfy are excluded outright — gradients against unattainable references degrade unrelated behavior long before they fail visibly. Second, dosage: an earlier variant that let the bank dominate sampling produced a specialist that had quietly lost the generalist's range. The schedule interacts with stopping (Section 6): once the bank is mastered, its continued rehearsal is a leading suspect for the post-peak generalization decline.

### 4.5 The native-body variant

Because the platform is frozen, the recipe does not care whose body the adapter serves. Dropping the codec and attaching the same adapters to the same decoder yields a native-body variant: the platform specializing on its own robot. Section 9 uses it as the experimental control that separates what the recipe does from what the embodiment gap does.

## 5 Results

All numbers below are success rate (%), mean, and 95th-percentile joint error (mm) over surviving episodes, under identical strict gates, same machine, per benchmark; survival rates from the second-referee engine are quoted where deployment robustness is at issue.

### 5.1 Zero-shot: how far the codec alone carries

With the codec and nothing else, the frozen platform stands, walks, and dances on a robot it has never seen. On a small set of easy in-regime clips its measured per-joint tracking error is comparable to the trained incumbent's — numerically even lower (6.1° versus 8.6° on a relaxed walk) — though we weight this narrowly: the sample is three easy clips, and qualitatively the zero-shot motion is visibly choppier than the incumbent's, a numeric-versus-visual divergence we did not resolve. What the numbers do support is the modest claim that matters: geometry survives the codec — the frozen prior produces recognizably correct motion on the new body before a single gradient step.

Robustness is another matter entirely: across a 200-clip battery the zero-shot controller survives roughly 75% of clips against the incumbent's 98%, and its failures concentrate exactly where the actuator gap lives — pelvis sink under deep flexion and single-support moments. The codec solves geometry; it cannot solve dynamics. That 23-point survival gap is the adapter's job description.

### 5.2 Adapted: the master comparison

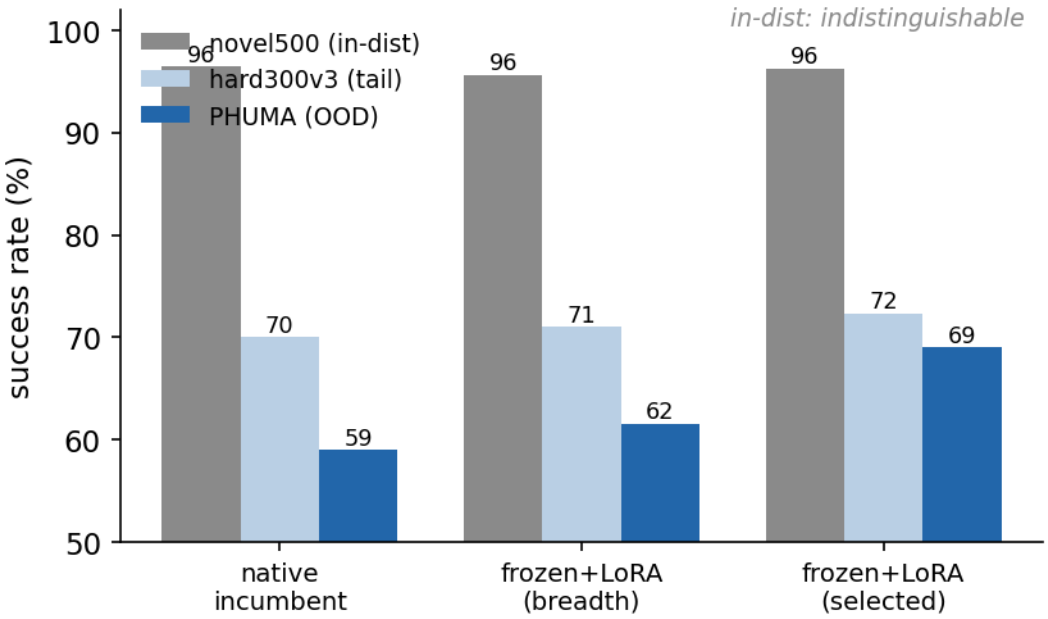


Figure 2: Success by benchmark: in-distribution parity, tail advantage, and the out-of-distribution reversal.

Table 1 summarizes the adapted transfer (its selected checkpoint) against the incumbent across all three benchmarks. Each cell reports success rate (%) / mean error (mm) / 95th-percentile error (mm) over surviving episodes; the final column is survival rate (%) on the out-of-distribution slice under the second-referee physics engine.

| model | novel500 (in-dist) | hard300v3 (tail) | PHUMA (OOD) | PHUMA survival |
|---|---|---|---|---|
| incumbent (native) | 96.4 / 33.6 / 43.8 | 70.0 / 40.6 / 56.0 | 59.0 / 42.6 / 67.4 | 87.4 |
| transfer (breadth only) | 95.6 / 33.7 / 42.9 | 71.0 / 40.0 / 56.4 | 61.6 / 42.9 / 60.9 | 89.4 |
| transfer (selected) | 96.2 / 33.1 / 42.9 | 72.3 / 39.9 / 55.8 | **69.0 / 41.7 / 60.6** | **90.7** |

Three readings. *In-distribution, the models are indistinguishable*: every X2 model lands near 96% success and 33 mm — differences within noise. A practitioner running only standard in-distribution evaluation would conclude the expensive frozen platform bought nothing. *The tail tells more*: the transfer leads on the curated hard set, and on the clips all models survive, its 95th-percentile error beats even the platform-on-native-body's tail. *The out-of-distribution column is the story*: ten

success points over the incumbent, with a better tail — and notably, the breadth phase alone, before any polish, already clears the incumbent. The frozen prior's generalization is present from the first phase; polish widens it. The second referee agrees where it matters, as the table's final column shows: survival on the same out-of-distribution slice runs 90.7% for the selected transfer against 87.4% for the incumbent.

### 5.3 Compute

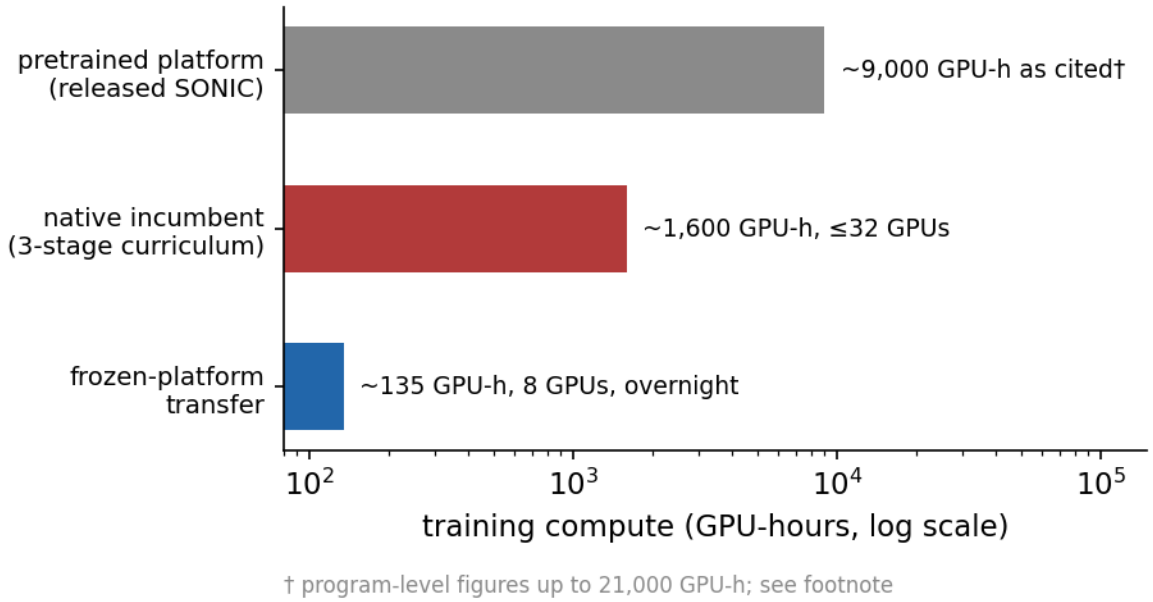


Figure 3: Training compute by tier (log scale).

The selected transfer checkpoint's entire training lineage — breadth, polish, and selection — cost roughly 135 GPU-hours on a single 8-GPU node: one node, overnight. The full experimental program, including every failed branch reported in this paper, stayed under 190. Against the platform's cited training compute this is roughly two percent; against the incumbent's ~1,600 GPU-hour curriculum it is roughly an eighth — spent to *beat* it out-of-distribution, not to match it.

## 6 The Training Arc: Slope, Peak, Decline

Out-of-distribution success is not a monotone reward for more training. Across the polish phase it climbs a measurable slope — roughly 2.7 success points per thousand iterations, with no plateau through the peak — and this slope, measured at every 500-iteration gate, is what justified continuing. Then it bends. Past the peak, four consecutive evaluations decline monotonically (69.0, 67.5, 65.8, 64.4) while in-distribution success stays flat within noise. The final, most-trained checkpoint is the *worst* out-of-distribution model of the phase — and, deceptively, its surviving-episode error is slightly *better*: precision up, coverage down, the classic signature of specialization eating generalization. Training-side metrics gave no warning; console reward was flat-to-down throughout and, under an adaptive sampler that reshuffles clip difficulty, unreliable in both directions. Only the gated out-of-distribution evaluation saw the bend — hence the stopping rule: halt after two consecutive declining gates, select the peak.

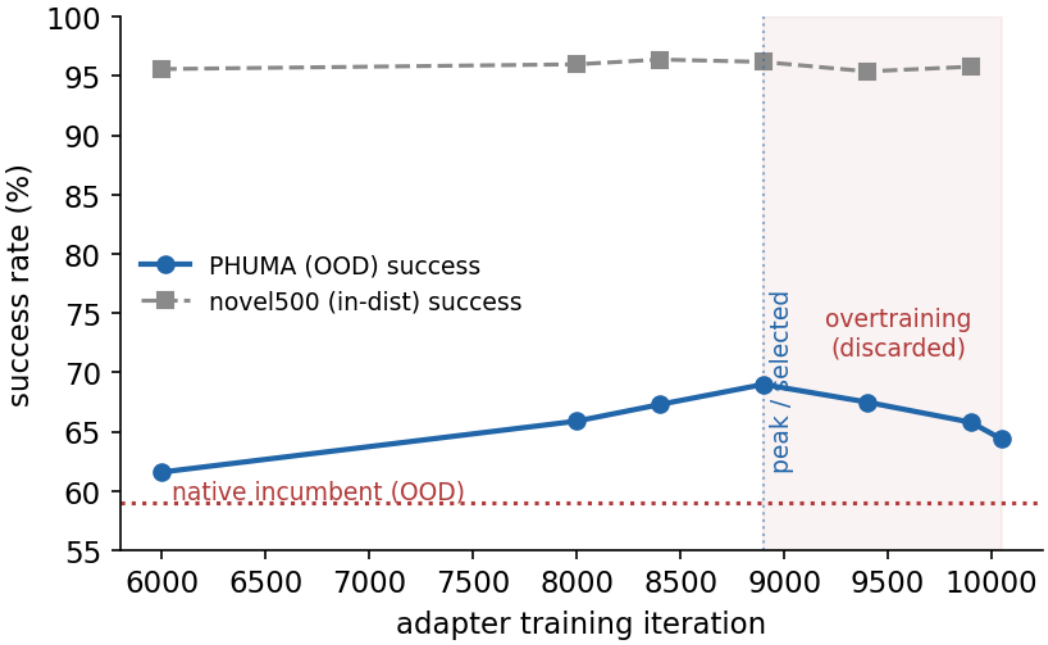


Figure 4: Out-of-distribution success across adapter training, with the selected peak and the discarded overtraining tail.

We report the suspected mechanisms and their status honestly. The timeline is consistent with a mastery-then-memorize account: the polish bank's in-distribution benefit saturates around the peak, after which its residual gradients — and possibly an auxiliary reward term that Section 8 proves unsatisfiable — push the small adapter toward per-clip specifics at generalization's expense. The two-arm ablation that would separate these mechanisms is specified but was outside this work's compute budget; we deliberately state the decline as an observed phenomenon with suspected causes, not an explained one.

## 7 The Embodiment Floor

Every X2 model in this paper — the incumbent trained natively for ~1,600 GPU-hours, and every checkpoint of the frozen-platform transfer — lands its mean joint error in the same narrow band, roughly 41–43 mm, on out-of-distribution content where the identical platform on its native body scores 32 mm. Training moves *coverage* (success rates swing by ten points across these models); it does not move the *floor*. Models sharing almost nothing — objective, trained parameter count (14.7M versus 245.7k), corpus era — converge within 1.5 mm.

We attribute the ~10 mm offset to embodiment cost — the actuator gap (roughly half the per-kilogram torque at ankles and waist) plus the retargeting itself — and support the attribution with converging probes: the transfer adapter's largest weight movements concentrate on exactly the ankle rows; simulated torque-saturation counters flag the same joints; and the retargeting pipeline's known compensations (the target's elbow must flex further to match hand positions on its differently proportioned arm) contribute measured tens of millimeters on specific clips. Three caveats accompany the claim: the floor bundles actuator and retargeting contributions we cannot yet decompose; "irreducible" means invariant across every lineage and budget we built, not provably

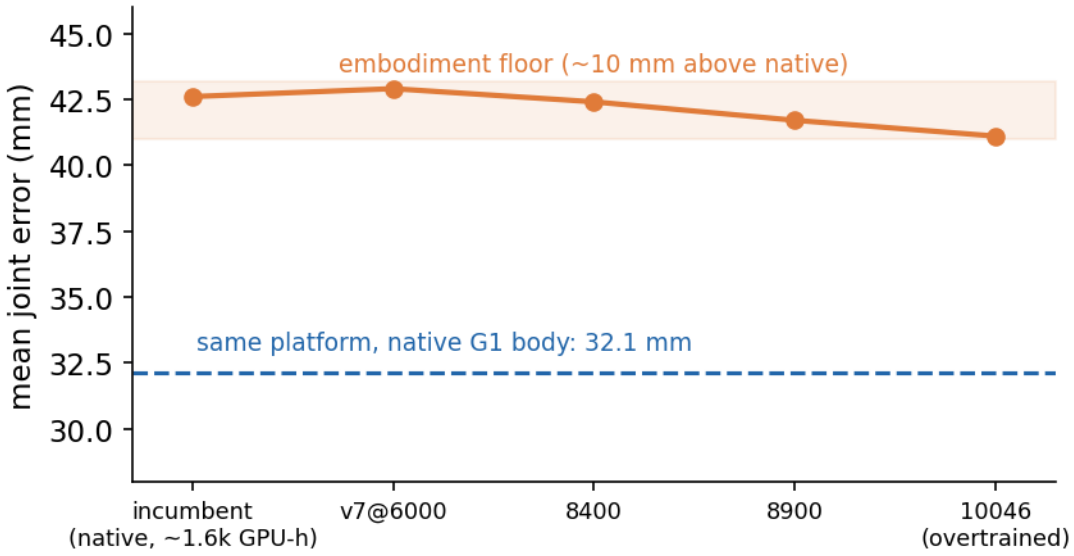


Figure 5: Mean error across independently trained X2 lineages, against the platform-on-native-body reference.

permanent; and means are computed over survivors — though the convergence *despite* different survivor sets strengthens rather than weakens the case, since better models keep harder clips alive and still land on the same mean.

## 8 What the Frozen Latent Cannot Carry

The freeze's second cost is informational. In the released training configuration, the tracked keypoints end at the wrist links and the reward composition contains no joint-space tracking term — a wrist roll, rotating about an axis through its own keypoint, moves nothing any reward can see — and on the transferred robot the wrists drift far from their references while everything else tracks. Four successive remedies failed: recalibrating the codec's wrist offsets (the drift is posture-dependent, not constant); surgically freezing the adapter's wrist output rows (the drift is distributed through shared features); and two attempts at an auxiliary wrist-tracking reward, narrow- and wide-kernel (the second had healthy gradients and still moved nothing).

The failures share one explanation, and it is measurable. We perturb a single reference joint by half a radian and measure how much of the change reaches the policy's commands. For a reward-visible joint (the elbow, as control) 73% passes through. For the wrist, 11% — barely above the run-to-run noise floor. The frozen encoder and its quantized bottleneck simply do not transmit absolute wrist pose; it was reward-invisible at platform training time, so the representation never learned to carry it — and no adapter downstream of the freeze, trained by any reward, can recover a signal that never arrives. This is the boundary of last-mile adaptation stated as a causal measurement rather than a conjecture. Practical mitigation is straightforward (deployments hold wrists at a fixed pose, which operator review found visually acceptable for demonstration content); the finding's value is the general lesson: *what the platform never learned to see, no downstream adaptation can restore.*

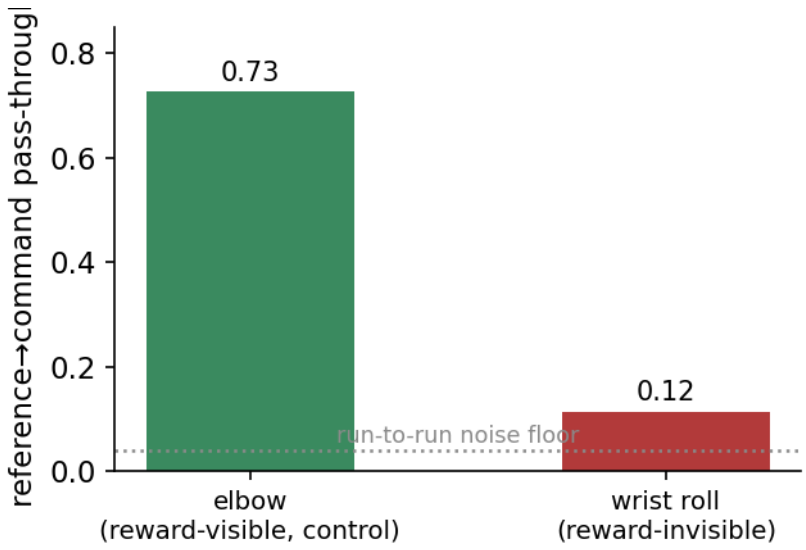


Figure 6: Reference-to-command pass-through for a reward-visible joint and a reward-invisible one.

## 9 The Same Method on the Platform's Own Body

Every result so far confounds two things: the recipe and the embodiment gap it crosses. To separate them, we run the recipe with the gap removed: same frozen platform, same adapters on the same decoder, but the body is the platform's own Unitree G1 — the codec becomes the identity, and every difference between reference and execution is attributable to the method alone.

The task is practical: the released controller fails a noticeable fraction of a curated demonstration bank — expressive dances the dominant failure family (the bank scores 93.6% under the release, with the failures concentrated in dance). We fine-tune on this bank with the polish-phase configuration, holding the platform's own broad benchmark out as a forgetting guard. If the recipe works for the reasons we claim, three things should happen: the bank should recover; the guard should hold; and the overtraining pathology of Section 6 should be absent or delayed, because every reference is attainable by this body in its native action space.

All three happen. The bank saturates — every previously failing demonstration motion recovered — within tens of GPU-hours. The guard not only holds but improves: the specialized controller tracks the platform's own broad benchmark slightly better than the released weights (98.6% / 23.0 mm against 98.2% / 25.7 mm at the best gate). And across a doubling and near-tripling of the budget, the guard stays flat within noise — no bend, none of the precision-up-coverage-down signature of the cross-embodiment run. Specialization on attainable, native-space content appears benign in a way that specialization across an embodiment gap is not — which is what Section 6's suspected mechanisms predict, though we report this as consistency, not proof: the two runs sit at different points of their schedules, and the separating ablations were outside budget.

Beyond its role as a control, the variant is useful in itself. It turns a released, frozen checkpoint into a robot-specific specialist without touching the release — the adapter is megabytes, the platform is shared, and the specialization is reversible by deleting a file. For platform vendors, this argues that shipping frozen weights does not freeze capability; for users, it is a recipe for making a generalist do a particular job without inheriting the risk of retraining it.

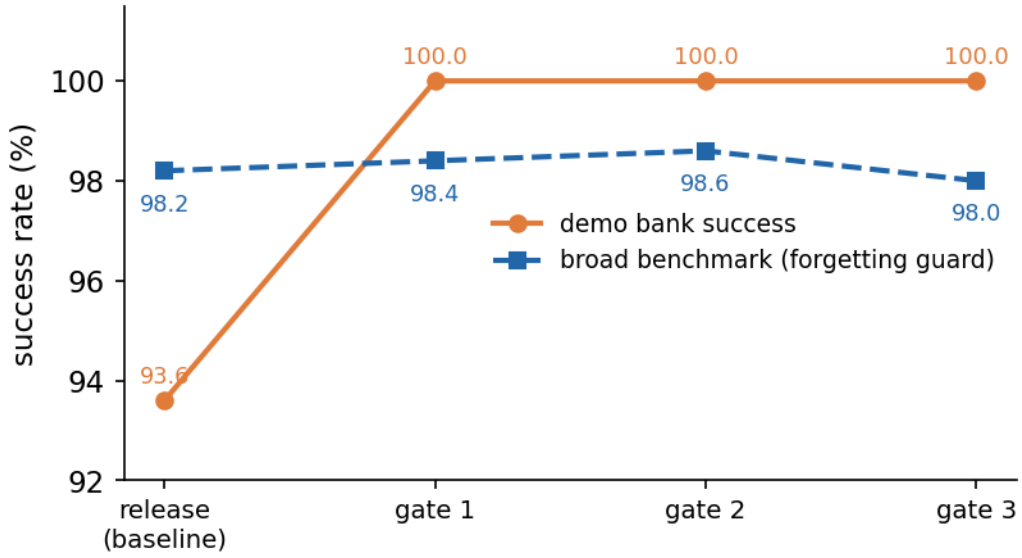


Figure 7: Native-body specialization: bank recovery with the forgetting guard holding across training gates.

## 10 Limitations

The foreign-body evidence is one robot pair, and a deliberately favorable one: the target was designed for skeletal compatibility with the source, and we report zero-shot survivability explicitly as a similarity-precondition measurement. Whether the frozen-platform recipe's advantages survive a morphologically distant pair — where Any2Any's learned alignment is the price of admission — is untested here. Our out-of-distribution benchmark served double duty during training as the stopping signal; the reported test numbers therefore come from a freshly drawn disjoint sample scored once. The overtraining decline and its mechanisms are observed and suspected, respectively — not ablated, by explicit budget decision. Our hardware-facing evidence for the selected checkpoint is simulation-side (dual-referee); physical-robot verification of the transferred controller follows the deployment process established for the incumbent and is ongoing. The embodiment floor is not yet decomposed into actuator and retargeting shares. And the wrist result, while causally grounded, is a single-platform measurement of a general claim.

## 11 Conclusion

A frozen foundation controller, wrapped in a closed-form codec and steered by a quarter-percent's worth of adapters, outperformed a purpose-built tracker on the robot that tracker was built for — on exactly the content neither had seen. The result reframes what a released checkpoint is: not a demo artifact welded to its authors' robot, but a transferable motion prior whose generalization — the most expensive thing in it — can be borrowed intact by any sufficiently similar body, overnight, on one node. The boundaries we mapped are as much the contribution as the win: what the body cannot do, no training buys back; what the platform never learned to see, no adaptation restores. Between those boundaries lies a wide, cheap, and now well-charted corridor. We release the transferred models, the codec and its calibration sidecar, the adapters, and the three-role evaluation protocol; video comparisons accompany this paper at the project page, https://sonic-

agibot-x2.github.io/sonic-transfer/.

## 12 Acknowledgments

The GEAR-SONIC team, for releasing the platform this work stands on.

## 13 Appendix (planned)

Full gate tables and per-checkpoint metrics; evaluation protocol and referee definitions; codec coefficient tables and sidecar format; compute reconstruction from run logs; evaluation-node bring-up notes.

## References

Hu, Edward J., Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. "LoRA: Low-Rank Adaptation of Large Language Models." *arXiv Preprint arXiv:2106.09685*.

Lee, others. 2025. "PHUMA: Physically-Grounded Humanoid Motion Dataset." *arXiv Preprint*.

Luo, Zhengyi, Ye Yuan, Tingwu Wang, Chenran Li, Fernando Castañeda, et al. 2026. "SONIC: Supersizing Motion Tracking for Natural Humanoid Whole-Body Control." *arXiv Preprint arXiv:2511.07820*.

Mentzer, Fabian, David Minnen, Eirikur Agustsson, and Michael Tschannen. 2023. "Finite Scalar Quantization: VQ-VAE Made Simple." *arXiv Preprint arXiv:2309.15505*.

Yang, Ming, Tao Yu, Feng Li, and Hua Chen. 2026. "Any2Any: Efficient Cross-Embodiment Transfer for Humanoid Whole-Body Tracking." *arXiv Preprint arXiv:2605.23733*.