Zero-Shot Cross-Embodiment Reuse of a Frozen Humanoid Motion Planner via Analytic Alignment and Closed-Loop-Selected Residual Heads

Sitarama Chekuri Patrick Rose (Ultimate Bots) Claude Fable 5 (Anthropic)

Draft v0.5 (compressed) — 2026-08-14

Abstract

Generative motion planners for humanoids are expensive to train and are increasingly available only as pretrained checkpoints whose training distribution cannot be reproduced: the source planner in this work was trained on a proprietary 350k-clip motion collection, of which only small public subsets exist — so further training on available data would degrade the model rather than extend it. Such assets can be run, but not meaningfully retrained. We show that a 2M-step motion planner trained entirely on one humanoid (Unitree G1) can drive a morphologically similar humanoid (AgiBot X2 Ultra) at deployment quality without a single gradient step on its weights. A purely analytic wrap — per-joint affine alignment fitted from a paired retargeting corpus, plus stride, height, and intent scaling — achieves zero-shot transfer that outperforms both planners natively trained for the target robot on closed-loop stability metrics. Small residual heads (83k parameters, decoder-side only) trained on 33k paired clips close most of the remaining kinematic gap. Our central finding is methodological: offline fit and closed-loop viability diverge sharply. The best-fitting head architectures destabilized the downstream whole-body tracker through train/deploy input mismatches invisible to validation metrics, and fitting the reference height exactly (1.3 mm) removed a floating-height convention the tracker depends on. Model selection by closed-loop rollout — never by reconstruction error — plus export-time channel masking and strength scaling turned one trained model into a searchable family whose winning configuration passed every gate: recovery steps at or below the native planners', reference smoother than the source wrap, and a 10x improvement over the deployed baseline on operator driving tapes. We further exploit the planner's template-conditioning interface — a documented feature of its architecture — as a repair channel for specific behaviors on the new embodiment: operator-recorded gait-initiation templates, harvested by jerk metrics and routed by a stateless standstill detector, fixed a core-native first-stride surge without any training. We report the full experimental ledger, including two closed-loop-falsified architectures (and a third rejected on validation), on one closely-matched embodiment pair, and characterize the morphological-similarity preconditions the method relies on.

1 Introduction

A robot demo needed a smooth slow walk. The target robot — an AgiBot X2 Ultra driven by a generative motion planner feeding a 50 Hz RL whole-body tracking policy (SONIC, "the tracker") — had two planners of its own, both undertrained relative to their reference architecture because GPU budget, not data, was the binding constraint: a deployed model trained early in the project on a small corpus (single 8xH100 node, roughly a quarter of the source planner's optimization schedule), and a fresh 32-GPU cloud run on a 129k-clip corpus, wound down at a tenth of that schedule when

the fleet's costs came due. Neither delivered the walk. The deployed planner produced acceptable but correction-heavy gait; the new planner, despite better numbers on every offline metric we measured, lost to it in closed loop where it mattered — on the seam sweeps and the primary driving tape (Section 3). Training to the source planner's maturity — roughly 10,000 H100-hours per Section 3 — was not an option we could afford to exercise.

We also had a third planner we could not train at all. The source robot's planner (Unitree G1) is the same architecture at identical tensor widths, trained by its authors to 2M optimization steps per model — roughly four to ten times the schedule our native chains reached — and available to us as a weights-only asset. We can retrain anything on the target robot's side; we can never retrain the source planner. This asset asymmetry defines the setting.

The bet this paper reports: the planner is kinematic — joint references, not torques — so the embodiment gap is retargeting, not dynamics, and the downstream tracker acts as a projection onto the target's feasible manifold. If so, a frozen planner should port across morphologically close humanoids with an analytic joint-space alignment ("the wrap") and at most small learned residuals ("the heads"). It was right, and cheaper than expected: the wrap alone beat both native planners zero-shot on the headline closed-loop metric — recovery steps {1,2,3,4} at walk 0.35 m/s versus 6 and 18–36, and roughly 4x fewer on operator driving tapes — and 83k parameters of per-frame residual heads, trained in minutes-scale runs on one workstation GPU, produced a configuration that passed every gate ({1,2,1}) with an operator verdict of ready to ship for the demo walk.

The contribution we consider most transferable is not the transfer result but the *selection discipline that produced it*: offline fit and closed-loop viability diverge, and the paper develops that thesis three times. The natively trained planner improved every offline metric and lost in closed loop. The two best-fitting head architectures destabilized the tracker through train/deploy input mismatches invisible to validation error — one made the system fall where nothing had fallen before. And fitting the reference height to 1.3 mm — offline perfection — removed a floating convention the tracker depends on. Every time we let an offline metric argue, it argued wrong.

Contributions: **(C1)** zero-shot planner transfer via analytic alignment, beating both native planners; **(C2)** a residual-head architecture study whose failures are reported as first-class results, with a deploy-distribution audit principle; **(C3)** closed-loop model selection with export-time masking and strength-scaling knobs, and a two-sided height ablation showing reference conventions are load-bearing; **(C4)** template-library injection — including operator-captured gait-initiation templates — as a training-free control surface of the frozen core; **(C5)** an operator-in-the-loop evaluation protocol with a published failure ledger, including two tape-parsing traps, one of which passed a quantitative gate and a human feel-test before an FK audit caught it.

We claim transfer for one closely matched embodiment pair, and we characterize the morphological preconditions the method relied on (Section 5). We do not claim "any to any."

2 Related Work

Cross-embodiment transfer for humanoids. The closest setting to ours is Any2Any (Yang et al. 2026), which transfers G1-pretrained whole-body tracking *policies* to new humanoids via analytic kinematic alignment (scattering matrix, axis decoupling, closed-chain Jacobian) followed by LoRA-scoped RL adaptation at roughly 1% of the original compute. We lean on two of its ablations: alignment-first is load-bearing (full fine-tuning without alignment loses to LoRA with it), and input-only adaptation is unstable. Our setting differs in what is transferred: we move the

kinematic planner and leave the target robot's own tracker in place. Because the planner emits references rather than torques, there is no dynamics gap to close with RL — the tracker projects the reference onto the target's feasible manifold — and the adaptation collapses to supervised regression on paired kinematics. This is why our transfer is zero-shot where Any2Any's requires reinforcement learning.

The planner line. The frozen core is a MotionBricks-family generative planner (Wang et al. 2026): a multihead VQ-VAE motion tokenizer with root and pose prediction models, conditioned on a short context window, velocity intent, and a template clip library. Both the source and target planners in this paper are this architecture at identical widths (Section 5); the paper's method exploits that identity rather than contributing to the architecture.

Motion retargeting. Classical and learned retargeting provides our loss vocabulary and failure taxonomy: FK-space and end-effector objectives (Aberman et al. 2020; Villegas et al. 2018), residual corrections over an analytic map with semantics-preserving regularizers (J. Zhang et al. 2023), skeleton-conditioned generalization (Lee et al. 2023), and the observation that retargeting quality gates downstream policy learning (He et al. 2024; Araújo et al. 2025). Our paired corpus is itself the product of an offline G1-to-X2 retargeting pipeline, and our corpus gates (including an IK branch-flip detector) follow this line.

Residual and adapter learning around frozen models. The heads are zero-initialized residuals over the analytic wrap, so at step zero the system is bit-exact the analytic baseline — the ControlNet initialization pattern (L. Zhang, Rao, and Agrawala 2023), also standard in residual policy learning (Silver et al. 2018). Staged fitting (affine terms before residuals) follows LP-FT (Kumar et al. 2022). Low-rank adaptation of the core itself (Hu et al. 2022) was planned as an escalation stage and never needed. Frozen-backbone reprogramming — learned input/output codecs around an untouched generative model — is the nearest published template: Time-LLM (Jin et al. 2024) for time series, PULSE-style distillation through a frozen core (Luo et al. 2024), and MotionGlot's per-embodiment codecs (Harithas and Sridhar 2025), though MotionGlot co-trains its backbone. To our knowledge no published work does exactly what we do here: a frozen generative motion prior with trained boundary codecs fitted on paired retargeting data, selected in closed loop.

Closed-loop training and selection. Our selection discipline — promotion by rollout metrics, never by reconstruction error — is the evaluation-side analogue of closed-loop supervised training methods such as CAT-K (Z. Zhang et al. 2025), which rolls a tokenized autoregressive model on its own outputs and supervises visited states. We adopted the weaker, cheaper form (closed-loop *selection* over offline-trained candidates) and found it sufficient; closed-loop *training* of the heads remains future work. The recurring observation that open-loop motion metrics fail to rank closed-loop behavior echoes exposure-bias findings in autoregressive generation and in trajectory forecasting (Z. Zhang et al. 2025, and references therein).

Positioning. In one sentence: prior work transfers policies with RL, or retargets motions offline, or adapts frozen backbones in other domains; we show that for kinematic motion planners feeding a competent tracker, cross-embodiment reuse needs only an analytic wrap plus tiny closed-loop-selected residual heads — and that the selection protocol, not the architecture, is the hard part.

3 The Case for Borrowing: Native Training and the Budget Wall

Before borrowing a foreign planner, we did the obvious thing: we trained our own. Twice. This section reports why that failed, because the failure mode is what motivates everything after it: not a

data wall but a *compute wall*. Cost ended training early; the deficit arrived as missing optimization steps; and the damage those missing steps do is invisible to offline metrics — it surfaces only in closed loop, as recovery steps.

The target robot's planner chain was first trained from scratch on a 33k-clip curated corpus, on a single 8xH100 node to roughly 500k steps per model (VQVAE and pose ~500k; root 315k) — a quarter of the source schedule on a quarter of the source's GPU count — then retrained at scale: a 129,363-clip corpus (97.4% SMPL-sourced) on a 32-GPU cloud fleet (Section 4). Budget constraints ended the big run at a fraction of the source planner's optimization schedule. The borrowed G1 chain has 2M steps on each of its three models; the native X2 chain reached 500k (VQVAE), 600k (root), and 200k (pose) before fleet wind-down.

On paper, the big run was a success: VQVAE reconstruction -52%, root target error -43%, pose top-5 at 95% versus the prior chain. Had we promoted on those numbers — an earlier reference-space evaluation round argued exactly that — we would have shipped a regression. In closed loop the big-run planner *lost to the years-old deployed planner* on the seam sweeps (18–36 recovery steps against 6) and on operator tape A (not universally: it won tape B, 24 against 34 — but the demo-critical regimes all sided with the old planner). Diagnosis: undertraining for its corpus — 200k pose steps against 129k clips leaves the model weakest precisely in the regime reconstruction metrics do not see: replan-seam continuity and gait rhythm. The thesis's first appearance, and not by a few percent — a factor-of-several loss hiding behind across-the-board offline gains.

The cost asymmetry is the headline. The source planner's reported training budget is 32 H100s for 13 days — ~**10,000 H100-hours**. Our big-run attempt burned ~1,500 H100-hours and produced a chain that loses in closed loop to a years-old baseline; pushing it to 2M-step maturity would have meant committing the remainder of that schedule with no guarantee the closed-loop crossover arrives before the budget does. The complete adaptation in this paper — the wrap (zero training) plus every head run, ablation, and gate — consumed ~**6 GPU-hours on one workstation RTX 5090**: roughly **three orders of magnitude less than replicating the source schedule**. (All accounting is in GPU-hours; no monetary figures.)

Two qualifications keep this honest. The adaptation is cheap only because a fully trained source planner *exists* and the target is morphologically close (Section 5) — the 10,000 H100-hours were spent by someone; the claim is about the *marginal* cost of serving a sibling embodiment. And the big run was not wasted: the same fleet trained the SONIC tracker used by every result here, and its 129k-clip retargeting corpus supplied the heads' paired training data. The budget wall stopped the planner, not the program.

Table (real numbers, to finalize): training budget vs closed-loop outcome.

chain	corpus	steps (VQ-VAE/root/pose)	GPU budget	closed-loop (recovery steps)
old deployed (heuristic-era)	—	—	—	6
native scratch	33k clips	~500k/315k/500k	single 8xH100 node	*(capture)*
native big-run	129k clips	500k/600k/200k	~1,500 H100-h	18–36
source G1 training (for reference)	G1-native corpus	2M/2M/2M	**~10,000 H100-h** (32xH100, 13 d)	—

chain	corpus	steps (VQ-VAE/root/pose)	GPU budget	closed-loop (recovery steps)
borrowed core + this paper's adaptation	33k pairs (heads only)	0 new core steps	**~6 RTX-5090-h**	**{1,2,1}**

4 Cloud Training Infrastructure: the 32-GPU Big Run on Nebius

The fleet behind Section 3's native attempt and the SONIC tracker used throughout: it is the evidence base for the cost-to-2M projection, and its operational lessons are underdocumented relative to how much run quality they determine.

Fleet. All cloud training ran on Nebius, whose sponsored GPU credits made the fleet possible: a dedicated tenancy, 4 nodes x 8 H100 on an InfiniBand fabric, provisioned through the Nebius API (the console's cluster preset was non-functional at the time; recipe in Appendix C), torchrun across the fabric, shared filesystem for checkpoints and data at rest. *(capture: topology figure.)*

The node-local compute rule. The share is for checkpoints and data at rest *only*: bulk compute against it — preprocessing, feature caching, aggressive log writes — degraded I/O for every node. Stage-to-local-NVMe, train locally, rsync back. Stated as a rule because we paid for it more than once before it stuck.

Throughput. ~217 optimizer steps/min at 32-GPU scale (pose at 512/GPU, effective batch 8192) — at that rate the 2M-step source schedule is a weeks-long commitment, the concrete content of Section 3's wall. The SONIC tracker reached its shipped 14k checkpoint in a day-class run on the same fleet. *(capture: per-leg durations and GPU-hours.)*

Table (to finalize): fleet legs, durations, and outcomes.

leg	hardware	duration	outcome
SONIC tracker (14k)	32 x H100 (4 nodes, IB)	~1-2 days	the shipped/gated tracker used throughout
VQVAE	~16 x H100	(capture)	500k steps, archived; same nodes then moved to pose
root model	16 x H100	(capture)	600k steps
pose model	16 x H100 (after VQVAE)	(capture)	200k steps at wind-down

Operational traps worth publishing. (1) *Rank-0-only checkpointing*: only rank 0 wrote checkpoints; scripts assuming per-node output silently produced nodes with no resume state — verify checkpoint provenance before trusting any resume. (2) *Restart transients fake regressions*: every restart produces sampler-metric excursions indistinguishable from data or curriculum regressions; never interpret one without a control log. Several alarming big-run "regressions" were restart artifacts. (3) *Reshuffle cadence vs metric windows*: the corpus reshuffles every 250 iterations, imprinting a sawtooth that shorter or incommensurate metric windows alias (we block on 1000);

per-episode error metrics are measured at termination and bias toward failure states — per-step reward was the only series trustworthy across runs.

Wind-down followed a fixed order — verify rsync of run state, copy out tracker checkpoints, pull artifacts local, *then* stop nodes, keeping one alive for share access until the archive verified. The alternative (stop first, discover a missing artifact after) is the most expensive mistake available at wind-down, and trap (1) makes it easy to commit. The planner legs ended at 500k/600k/200k as reported; the infrastructure did its job — the budget wall was a schedule fact, not an operations failure.

5 System Setting

The planner. Both robots run the same MotionBricks-family architecture: multihead VQ-VAE tokenizer (4 frames/token), root model conditioned on a 4-start/4-end constraint window, pose model, all at 30 fps in a shared skeletal representation; a template mode conditions generation on clips from a library that is *data*, loaded beside the weights (Section 9 exploits this). The planner is kinematic — joint references and root trajectories, never torques.

The contract. The deployed graph takes a 4-frame qpos context plus a 4-vector velocity intent and returns a 64-frame plan. Two contract facts shaped the program: the context is *forward-looking* (the daemon supplies the next 4 frames of the served plan, not history), and only roughly the first 16 frames of any plan are ever served before a replan. An evaluation harness that ignored the second fact fabricated defects and misled a review round (Section 6); every evaluation here replicates the 16-frame serve cadence.

The target stack. A daemon serves plan frames (30 fps resampled to 50 Hz); SONIC, a 50 Hz RL whole-body tracking policy (shipped 14k checkpoint, Section 4), converts the reference into torques. The tracker is best-effort by construction — it *projects* the reference onto the robot's feasible manifold. That projection property is why a kinematic-planner transfer can work at all, and Section 8 shows why "improving" the reference can therefore backfire.

The morphological preconditions — explicit and partly deliberate (the target's skeleton class was built for weight reuse with the source's): identical 34-bone tree and representation widths (414-dim global / 413-dim local; the source spends its last two bone slots on dummy hand-roll bones where the target has real head joints); naming-level DOF correspondence (dropping the target's two head joints, owned downstream by a head-look overlay, leaves an exact 29-to-29 map by name — the paired-data fit of Section 6 confirmed no hidden permutations and no sign flips beyond one wrist-axis naming swap); the only dimensional divergence at the qpos boundary (36 vs 38) lives in a weightless converter, not a checkpoint tensor; and matched intent semantics up to scale. We report these as preconditions, not incidentals — they are what the cost model depends on, and Section 11 names the in-house embodiment (missing joints, sign flips, half the range of motion) on which we expect the wrap to strain.

Evaluation. The headline closed-loop metric is *recovery steps* — unscheduled corrective steps the tracker takes — counted by a daemon-faithful seam-sweep harness driving the full planner-daemon-tracker stack under scripted intents; recorded operator driving sessions ("tapes") replayed at original timing serve as regression artifacts. All results are simulation; hardware deployment is future work (Section 11).

6 Stage S0: Analytic Alignment

Construction. The wrap is a pair of analytic maps around the frozen core: PhiEncode takes target qpos (38) to source qpos (36) for the context; PhiDecode inverts it on the 64-frame plan. *Arms*: per-joint affines from the robots' existing retargeting calibration, independently reproduced by an empirical fit on 60 paired executed clips (R^2 approaching 1.0 — a change of coordinates). *Legs and waist*: fitted affines from the same data, and here the affine story is weak (R^2 0.11–0.89; the target's legs come from IK against different link lengths) — this measured residual is exactly the budget Section 7's heads absorb, sized before anything was trained. *Wrist*: a vendor naming-and-sign swap. *Root*: stride (xy) scaling by 0.8235, a height affine, heading passthrough, anchored at the last context frame. *Intent*: linear velocity channels scaled up by 1/0.8235 on encode — a bug caught by a directional smoke test (0.4 m/s commanded, 0.31 realized) that reconstruction parity would never flag. All three stages fuse into one ONNX graph presenting the target planner's exact contract; daemon, harness, and robot stack run it unchanged.

Mode routing, and a trap. The source library's inherited mode labels were wrong — kinematic labeling showed the "locomotion" modes were idle variants, so early configurations drove every gait from an idle template. An intent-aware in-graph router fixed it. Lesson: never inherit mode semantics from comments; label clip libraries from their kinematics.

Zero-shot results. {1,2,3,4} recovery steps across seeds at walk 0.35 versus the deployed planner's 6 and the big-run's 18–36; on operator tapes, 15-vs-58 (A) and 8-vs-34 (B); replan latency at parity (the wrap is computationally free); and crouch walking at 0.3–0.5 m/s arrives gratis from the source library — a capability neither native planner delivered.

Table 1 (real numbers): zero-shot closed-loop results.

metric (walk 0.35, sim)	old native	new native	**S0 wrap**
seam-sweep recovery steps (3–4 seeds)	6	18–36	**{1,2,3,4}**
operator tape A extra steps	58	—	**15**
operator tape B extra steps	34	—	**8**
replan latency p50 (ms)	44.8	—	**46.0**
crouch walking	no	no	**yes (0.3–0.5 m/s)**

Harness fidelity. An early visual review of the wrapped planner reported four defects — splayed turn legs, forward lean, a collapsing run, a crouch that did not crouch. All four were artifacts of a harness that consumed 48 frames per plan (the autoregressive tail) instead of the daemon's 16-frame serve. Re-rendered at deployment cadence, the defects vanished; the retraction is part of the ledger, and 16-frame serving is now load-bearing in every evaluation tool. Side-by-side with the source robot running the native core, the operator judged the wrapped output to match the source's motion — the wrap neither adds nor loses style, including the source's own flaws (its in-place turns slide on both bodies).

Attribution by A/B isolation. High-speed commands fail through the wrap — and the right question is whose failure it is. We ran the same conditions through (A) the wrapped planner with the wrap inside the autoregressive loop, and (B) the native source core decoded one-shot with the same map applied once, both tracked by the same tracker. B failed cell-for-cell like A: the source core's own speed envelope collapses at high asks natively, and clean fast references fall at the tracker's coverage boundary on both paths. Speed failures are core envelope plus tracker coverage,

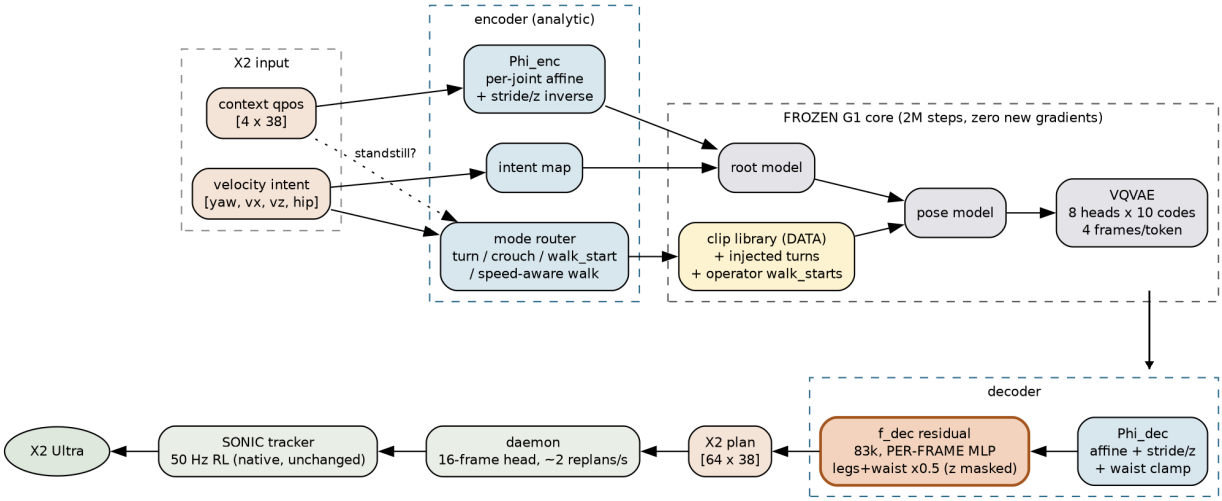

Figure 1: **F1 — System design.** The analytic encoder wrap and intent map feed the frozen G1 core (2M steps, zero new gradients); the mode router selects templates from the clip library, which is *data* — extended with injected turn modes and operator-captured walk-start templates. The decoder applies the analytic inverse plus the 83k-parameter per-frame residual head (legs+waist at half strength, height channel masked), and the plan flows through the unchanged daemon and native SONIC tracker.

not transfer error — and the isolation pattern (in-loop versus one-shot application of the same map) is, we think, the clean instrument for making such attributions.

7 Stage S1: Residual Heads — an Architecture Ladder

Data. The heads train on 33,204 frame-exact source/target qpos pairs from the retargeting pipeline (alignment verified at lag 0; arm R^2 0.996 median through the analytic affines). We got the corpus gates wrong twice — a variance-normalized R^2 gate cut fine near-static-arm clips; a jerk ceiling set below the corpus median silently selected a low-jerk subcorpus — before the final gate (arm RMSE, calibrated jerk, knee-flip drop) settled at ~31k clips. The knee-flip detector validated itself: from pair statistics alone it flagged 821 clips, rediscovering a known list of ~808 IK branch-flips from a prior corpus investigation. All training losses are computed at the source planner's native token granularity — 4-frame chunks, never per-frame — matching MotionBricks' own training structure (Wang et al. 2026), with source-trainer-style segment sampling (random clips and offsets; variable 24–64-frame segments for stage 5, so the deployed 64-frame plan length is in-distribution). The granularity constraint binds the *loss*; the function class is the variable under study.

The ladder. Five stages, all zero-initialized residuals over the wrap (bit-exact S0 at step zero), all trained on one workstation GPU: (1) *linear probe*, ~40k params, affine terms only — beat the per-joint analytic fit, gate to proceed; (2) *windowed MLP*, 817k, on non-overlapping 4-frame windows — best validation numbers to that point; (3) *temporal conv*, 852k, stationary over frames, built to fix stage 2's failure — matched its fit; (4) *per-frame MLP*, 83k, no temporal context, position-only, decoder-side, masked to legs, waist, and height — best fit of all five (legs 2.54°, waist 3.02°, height 1.32 mm); (5) *variable-segment conv*, 382k — lower training loss than stage 4, *worse* validation. Not gated.

Table 2 (real numbers): offline fit vs closed-loop viability.

heads	params	val legs MAE	val z MAE	recovery steps	falls
none (analytic)	—	5.61°	7.8 mm	{1,2,3}	0
windowed MLP[1]	817k	2.73°	1.4 mm	{21,25,32}	0
temporal conv	852k	2.64°	1.4 mm	—	**3/3**
per-frame (ours)	**83k**	**2.54°**	**1.3 mm**	{8,16,19}→ see §6	0
variable-seg conv	382k	2.73°	1.4 mm	not gated	—

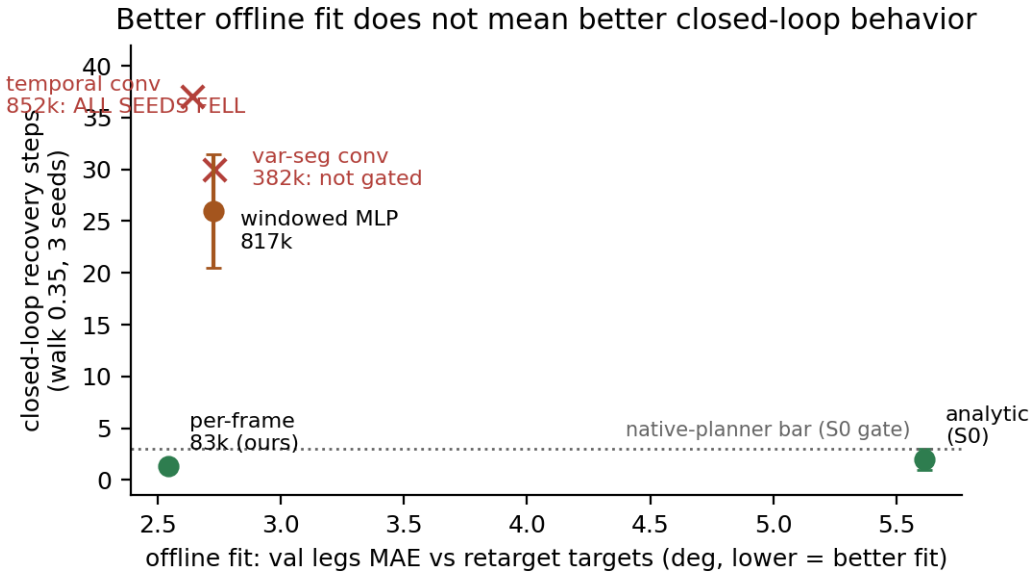

Figure 2: **F2 — Offline fit vs closed-loop viability, per head architecture.** Both large temporal models fit the targets well and fail closed-loop (one destabilized the tracker outright); the 83k per-frame model is simultaneously the best fit and the only architecture at the native bar.

Failure one: the windowed MLP. Offline best to that point; {21,25,32} recovery steps against S0's {1,2,3}. The sweep's telemetry located the mechanism: mid-plan acceleration at 3x baseline — jitter *inside* the plan, not at replan boundaries. On non-overlapping windows a frame's correction depends on its window slot, so the residual jumps at every boundary: a structural 7.5 Hz discontinuity. Per-window validation MAE is blind to it by construction — every validation window is scored independently.

Failure two: the temporal conv. Built to eliminate window slots; matched the MLP on validation; all three seeds *fell* — the program's first falls (seam acceleration 6x baseline, reference jumps of 3.3 rad). Post-mortem: three faces of one train/deploy input mismatch — the encoder head's 9-frame

[1]Val MAE from the corrected-corpus run; recovery steps from the architecture-identical run 1 (accidental low-jerk subcorpus) — the two runs share the window-slot defect by construction.

receptive field exceeds the deployed 4-frame context, putting every context frame in replicate-padded edge territory training never showed, which corrupts the input the *frozen generative core* conditions on; finite-difference velocity features, fitted on mocap-smooth data, amplify core-output noise at deploy; and the padding effects land exactly at replan handoffs.

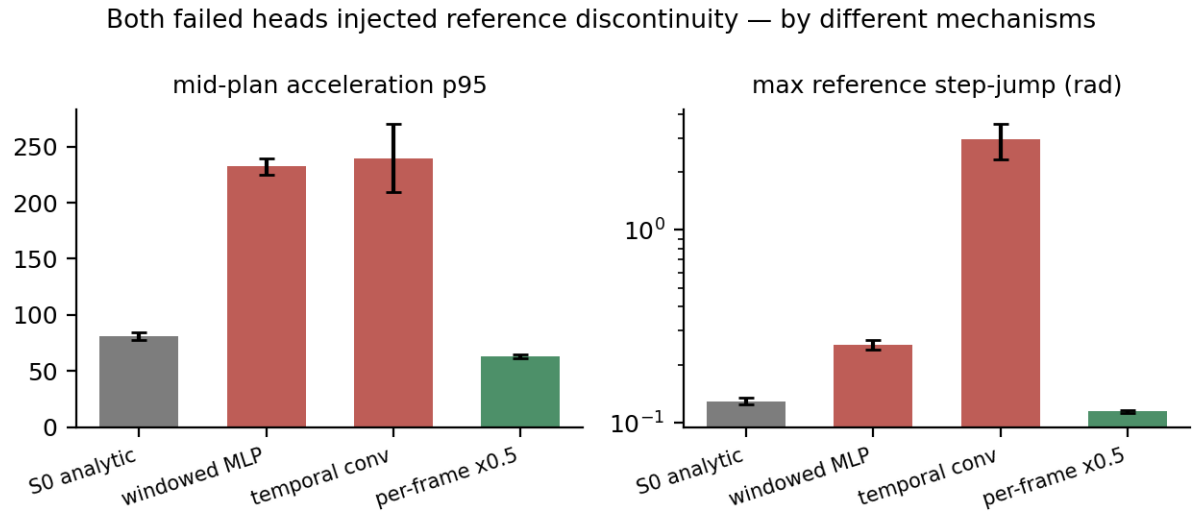

Figure 3: **F3 — The two failure mechanisms in the served reference.** The windowed MLP raises mid-plan acceleration ~3x (window-slot discontinuity); the temporal conv raises the maximum reference step-jump ~25x at replan seams (context-edge corruption upstream of the frozen core). The per-frame heads produce a reference smoother than the analytic baseline.

The audit principle, and the winner. With a frozen core in the loop, the heads' deploy-time input distribution is defined by the *pipeline*, not the dataset; every design choice must be audited against "what does this see at deploy that training never showed it." Stage 4 is that principle applied as architecture — per-frame (no slots, no edges), position-only (no velocity amplifiers), decoder-only (the core's inputs stay bit-identical to gated S0, deleting the fall class outright) — and it has the *best* offline fit at a tenth the parameters. Stage 5 confirms it from the other side: added capacity lowered training loss and worsened validation. The residual the wrap leaves behind is pose-conditioned geometry; temporal context buys only failure modes, and training-loss curves ranked the ladder exactly backwards.

8 Closed-Loop Selection and the Height Story

Section 7 selected an architecture; this section selects a configuration — where the method lives. Every candidate runs a gate battery: multi-seed seam sweeps (recovery steps + reference telemetry), operator-tape replays, an FK stance audit, and an operator feel-test as final acceptance. Offline validation is recorded but never promotes.

The best-fit model fails its first gate — the float is load-bearing. The per-frame heads at full strength scored {8,16,19} recovery steps against analytic S0's {1,2,3}, despite the program's best validation table and a reference *smoother* than S0. The clue was quantitative: {8,16,19} matches the {11,12,15} measured earlier when we "fixed" the wrap's height convention from the serve side. The wrapped reference floats — stance feet above the native convention — and the heads

fit ground-truth height to 1.3 mm, *removing* the float. The same ablation, run from both directions (analytic serve-side offset; learned height head), returned the same signature: recovery steps roughly tripled. Offline contact statistics call the float an error; the tracker calls it a convention it depends on. (Mechanistic hypotheses remain untested — Section 11 — but the selection consequence is unambiguous.)

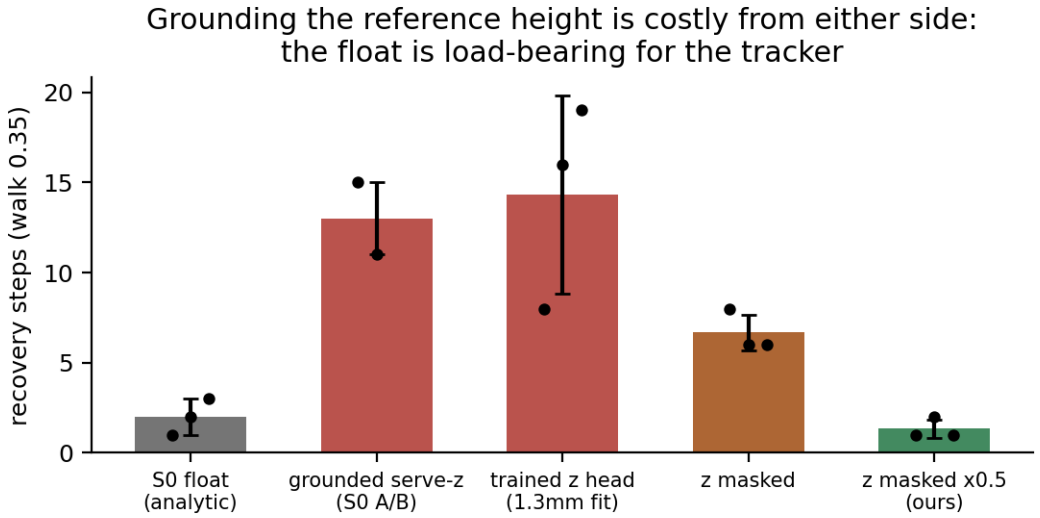

Figure 4: **F4 — The height ablation, two-sided.** Grounding the reference height — whether by offsetting the served root height (S0-era A/B) or by a z head fitting the retarget targets to 1.3 mm — costs several times the recovery steps of keeping the floating convention. Masking the z channel and halving the joint corrections recovers the stability crown.

Export-time knobs make one model a family. The heads' output passes through a channel mask and a scalar strength multiplier that are buffer edits at export — no retraining. Masking the height channel recovered most of the gap ({8,6,6}): the residual damage was the leg corrections themselves, since ground-truth leg angles encode ground-truth (grounded) foot height, so exact legs under a floated root are an internally inconsistent reference. Halving the leg and waist corrections resolved that tension: **height masked, x0.5 strength scores {1,2,1}** — at or below analytic S0's {1,2,3} — while keeping most of the style gain, with a smoother reference than S0 (acceleration p95 61–65 vs 78–84; step-jump 0.112–0.116 vs 0.124–0.134). One trained model, a searchable configuration family, and the search ran in closed loop at rollout cost only.

Table 3 (real numbers): the winning configuration ladder.

config (per-frame heads)	recovery steps	acc_mid p95	jstep max
full z + legs + waist	{8,16,19}	53–60	0.13–0.16
z masked	{8,6,6}	66–70	0.13–0.15
z masked, ×0.5 strength	**{1,2,1}**	**61–65**	**0.11–0.12**
source wrap (S0)	{1,2,3}	78–84	0.12–0.13

What it bought. On the FK audit (method-validated: it reads the old planner at exactly its

known 7.68 cm ±0.2 stance convention), the shipped configuration walks at 8.87 cm ±0.4 cm — stance wobble from the operator-flagged ±1.5 cm to the ±0.4 cm target band *through leg corrections alone*, float from +1.8 to +1.2 cm with the height channel masked (the remainder deliberately kept — Section 11). Tape results in Section 10.

The selection ledger. Offline fit ordered the head candidates nearly inseparably and preferred full strength over half. Closed loop ordered them {per-frame » windowed MLP » temporal conv} and preferred half strength by a factor of several. Promoting by validation would have shipped, in order: a 10x recovery-step regression, a falling robot, and a floating-convention violation. Offline metrics here were not merely uninformative — they were anti-predictive at every decision that counted.

9 Template Injection and Routing

The frozen core's template mode conditions generation on clips from a library that ships beside the weights — a documented feature of the source architecture, not a discovery of ours. Our contribution is using it as a *repair channel*: the library is data, so extending it does not violate the frozen-weights constraint, and behaviors the core cannot produce can be grafted in from recordings.

The motivating gap is slow walking, and it is *source-native*, not a transfer artifact: even on its own robot the core authors a surging first stride from standstill. The operator — who had once trained a target-native model on extra-slow walk data precisely because it started more gently — identified the surge immediately and confirmed it on the native reference. Intent-side rate limiting cannot remove it (a cold-start ramp A/B at two time constants left the first strides unchanged while making launch sluggish): the surge is authored by the generative core from standstill context, reflecting its training corpus. The demo requirement was smooth slow walking below 0.6 m/s, so we closed the corpus's gap through the template channel.

Turn templates. The core's in-place turns slide — source-native style, confirmed on the native reference (Section 6). We encoded the target's proven heuristic-era turn primitives through the wrap into source joint space as library modes 15/16, routed by a yaw-dominant intent test. They step in direct drive; through the deployed daemon path the slide persists (suspect: context filtering and 30-to-50 Hz resampling flattening the weight-shift steps). In-place turns remain the accepted, documented inability of the shipped configuration.

Gait-initiation templates, operator-captured. The fix inverts the ramp's bad trade: pay the sluggishness *once*. The operator drove a capture session under a deliberately long ramp constant; we harvested it by jerk metrics — 15 start attempts, 4 rated GOOD (initiation-to-steady jerk ratio ~1.0, height dip under 10 mm), one left-lead and one right-lead — into library modes 17/18. Routing is stateless and in-graph: context xy displacement under 1 cm plus a forward command routes to walk_start; once moving, the condition dies naturally. Gate cost is small ({4,3,3} vs {1,2,1}, the +2 at stop/start events where the counter cannot tell a stumble from the recorded weight shift; tape B neutral at 6=6). Operator, on the result: "all good… this is real."

Style follows the template. At higher walk speeds the system produced hurried slow-walk strides — the walk mode rode a slow template regardless of commanded speed. A speed-aware template switch (above 0.55 m/s, route to the fast-walk clip) fixed the stride character with zero training. Template selection is a per-intent style knob on a frozen core, and it transfers verbatim.

Two tape-parsing traps, reported in full. (1) The capture tape records the *served 50 Hz stream*, not 30 fps planner frames; the first build silently produced 1.67x slow-motion clips — caught on

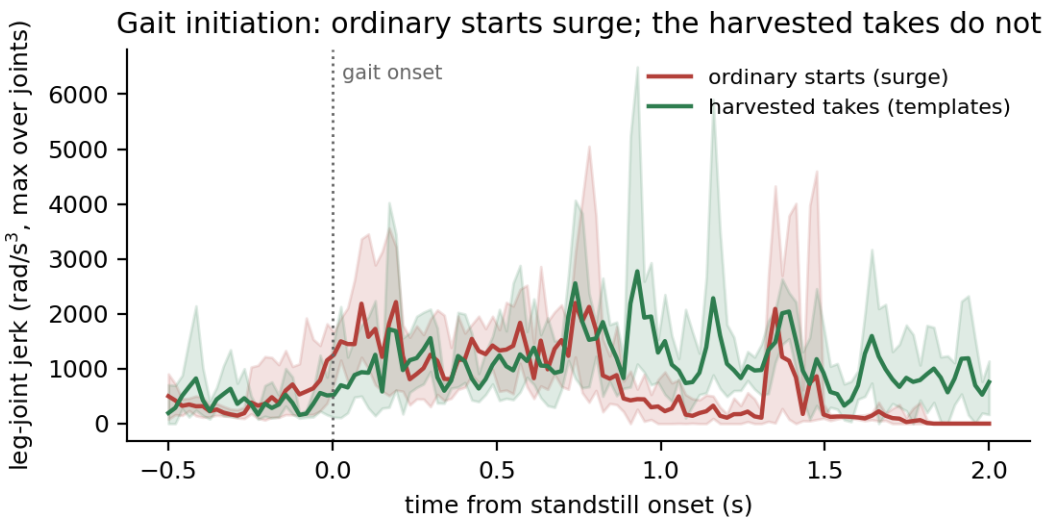

Figure 5: **F5 — Gait initiation in the capture session.** Leg-joint jerk around standstill onset, averaged over the session's ordinary starts (surging) versus the four harvested takes (initiation jerk at or below their own steady-state level); shaded bands are min–max. The harvested takes became the walk-start library templates.

review, rebuilt with explicit rate detection. (2) The tape stores its root quaternion in XYZW order; the planner convention is WXYZ. Reading one as the other scrambles root orientation while leaving joint columns coincidentally aligned — and the first walk_start build shipped wrong-orientation clips that *passed the quantitative gate and an operator feel-test*. Only the FK stance audit, which reads absolute foot heights and returned an absurd number, caught it. Rebuilt from the archived session, re-gated, re-confirmed. Lessons: one shared parser for every tape consumer; and gates that measure relative stability can certify a reference that is *wrong* in ways the tracker absorbs — absolute-geometry audits belong in the battery.

Limits of the library. Injection cannot exceed the core's data floor: commanded crouch depth saturates near the corpus's lowest hip height, and the crouch route rides a crouch-*run* template whose flight phases appear as 29% airborne reference frames (a calmer clip is queued). The library is a control surface over behaviors the core can already express, not a mechanism for synthesizing new ones.

10 Results

Table 4 carries the headline numbers; this section reads them.

Table 4 (real numbers): final system vs baselines.

	old native	S0 wrap	**final (ours)**
seam recovery steps	6	{1,2,3}	**{1,2,1}**
tape A (350 s, 84 stops)	58 extra	fell @96 s	**full tape, 26 extra**

	old native	S0 wrap	**final (ours)**
tape B (103 s, 16 stops)	34 extra	8 extra	**6 extra**
stance wobble (p10–p90)	±0.2 cm	±1.5 cm	**±0.4 cm**
stance float vs native	0	+1.8 cm	**+1.2 cm**
reference jitter (acc_mid p95)	—	78–84	**61–65**
operator verdict	—	"impressive zero-shot"	**"ready to ship"**

Stability. The final configuration (per-frame heads, height masked, half strength, plus walk-start templates) holds the program's recovery-step crown: {1,2,1} on the seam sweep versus the deployed planner's 6 and S0's {1,2,3} (the template-extended build gates {4,3,3}, the difference concentrated at stop/start events where the counter reads the recorded weight shift as a step). On operator tapes — the closest instrument to deployment — it survives the full 350 s of tape A where S0 fell at 96 s, at roughly half S0's per-second extra-step rate, and scores 6 on tape B against S0's 8 and the deployed planner's 34: an order of magnitude fewer corrections, and a fall converted into full-tape survival.

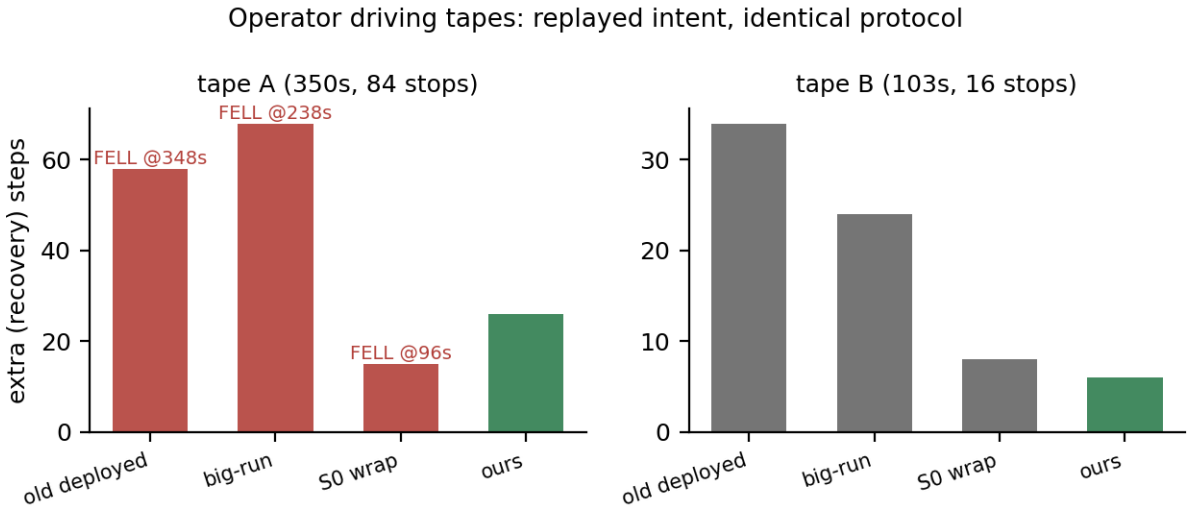

Figure 6: **F6 — Operator driving tapes, replayed under identical protocol.** On tape A the S0 wrap falls at 96 s; the final configuration survives the full 350 s with roughly half the per-second correction rate, and improves on tape B as well.

Reference quality and cost. The final reference is smoother than the wrap's (mid-plan acceleration p95 61–65 vs 78–84; step-jump 0.112–0.116 vs 0.124–0.134) and better grounded: stance wobble ±0.4 cm vs ±1.5 (native: ±0.2), float +1.8 → +1.2 cm — via leg corrections alone, height channel masked (Section 8). Stop poses moved as a side effect (knee at cut 22–42° → 21–35°, toward the native ~16° style), which may absorb part of a queued stop-logic retune. Replan latency is at parity (p50 44.8–46.0 ms) — the wrap and heads are computationally free — and the whole adaptation cost ~6 workstation-GPU hours against the source schedule's ~10,000 H100-hours (Section 3).

Capabilities. The capability matrix is deliberately honest. Slow walk at 0.3 m/s — the demo

requirement that started the program — is operator-verdict ready to ship, with one scoped caveat: the residual foot taps and recovery steps that remain concentrate almost entirely in tight moving turns, not in straight-line gait (the operator's estimate from the final drive was smooth "97% of the time", with the exceptions in turns). We treat tight-turn behavior as the top open item, distinct from the in-place slide below. Crouch walking at 0.3–0.5 m/s is a new capability neither native planner had, inherited from the source library, with a known depth floor. In-place turns slide (accepted, documented inability; the injected step-turn templates work only in direct drive). High-speed gaits are out of envelope by the A/B isolation of Section 6 — a bound belonging to the source core's corpus and the tracker's coverage, not to the transfer.

Operator verdicts are acceptance data here; the load-bearing ones: "pretty impressive for a zero-shot port" (S0 first drive); "walk is much better" (winning heads); "all good... this is real" (fixed walk-start build); and, on the shipping configuration, that the 0.3 m/s demo walk "looks ready to ship."

The thesis, third instance. Every column the final system wins was selected *against* an offline metric's advice at least once: the planner (Section 3), the architecture (Section 7), the configuration (Section 8). Offline fit answers "does the head reproduce the targets"; closed-loop rollout answers "can the tracker stand on the result." The method is the discipline of never confusing the two questions.

10.1 Qualitative comparison: source and target, same commands

Kinematic reference playback, source robot (left columns, native G1 core) vs target robot (right columns, wrapped planner with heads), identical velocity commands; two gait phases per cell.

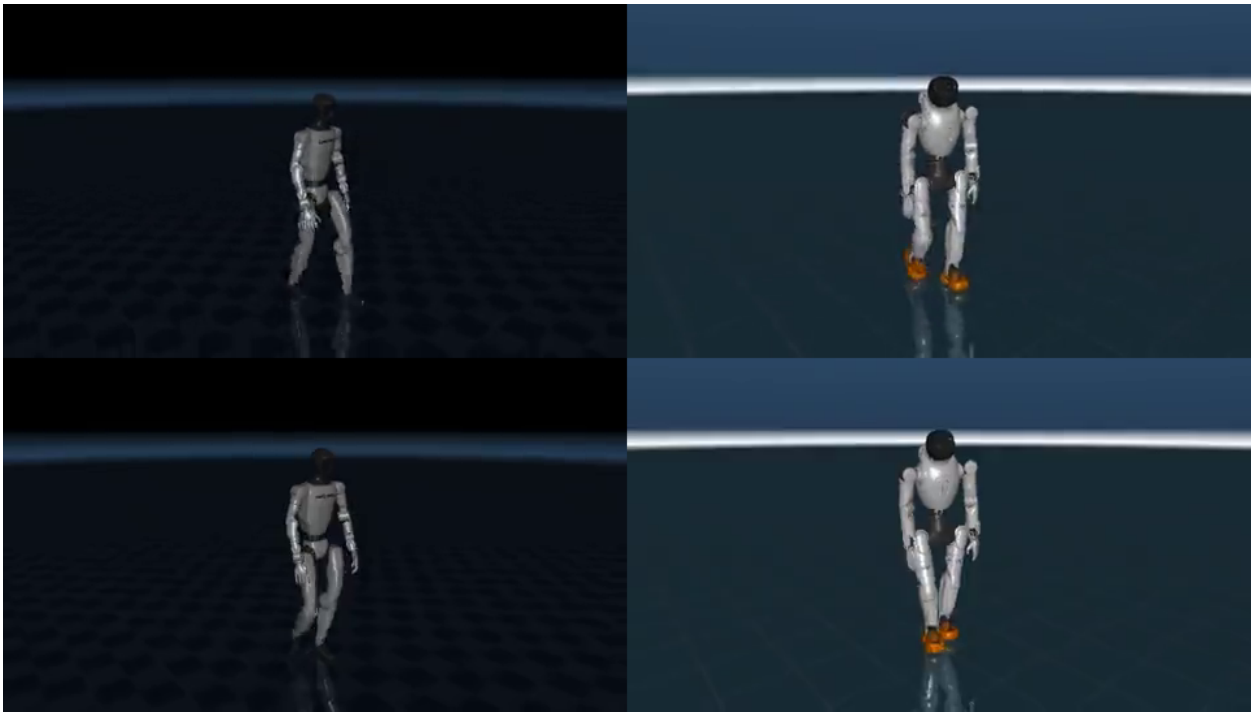

Figure 7: **F9a — Slow walk, 0.3 m/s.** The demo-critical gait: the target reproduces the source's stride character through the wrap; this is the configuration the operator signed off as demo-ready.

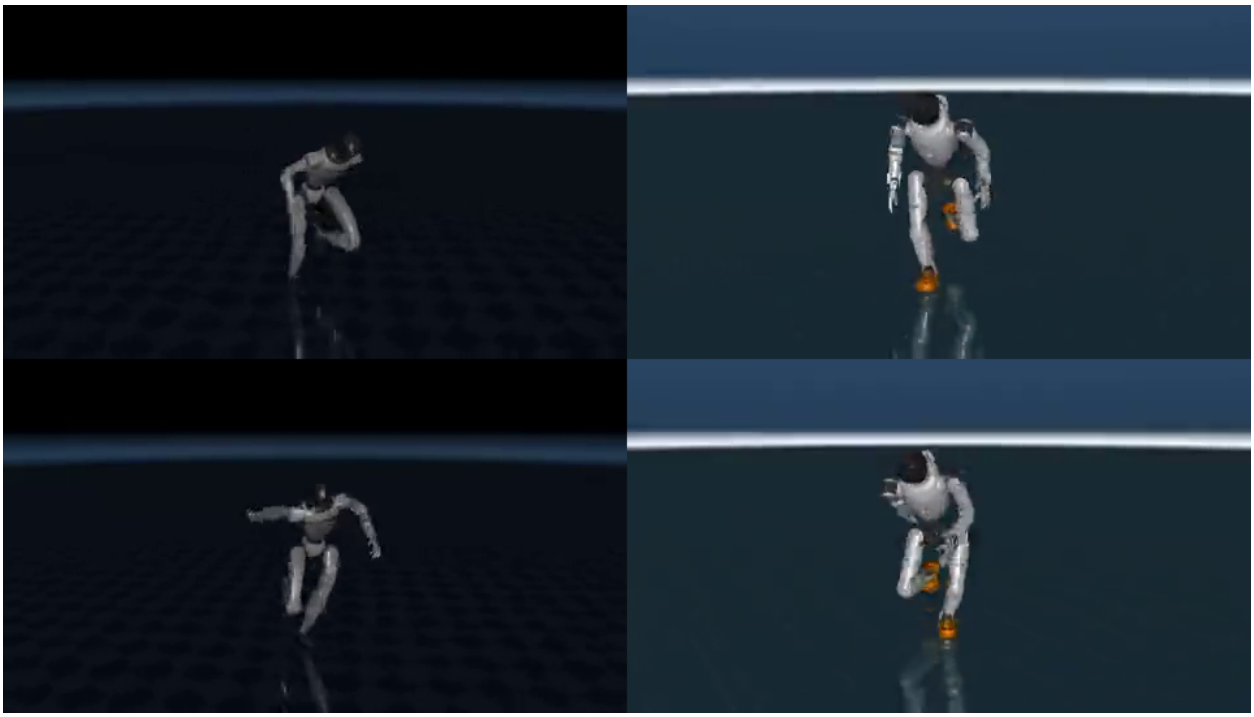

Figure 8: **F9b — Run, 3.0 m/s command.** Out-of-envelope stress case (the tracker's coverage, not a transfer defect — §5); stride style follows the fast-walk template on both sides.

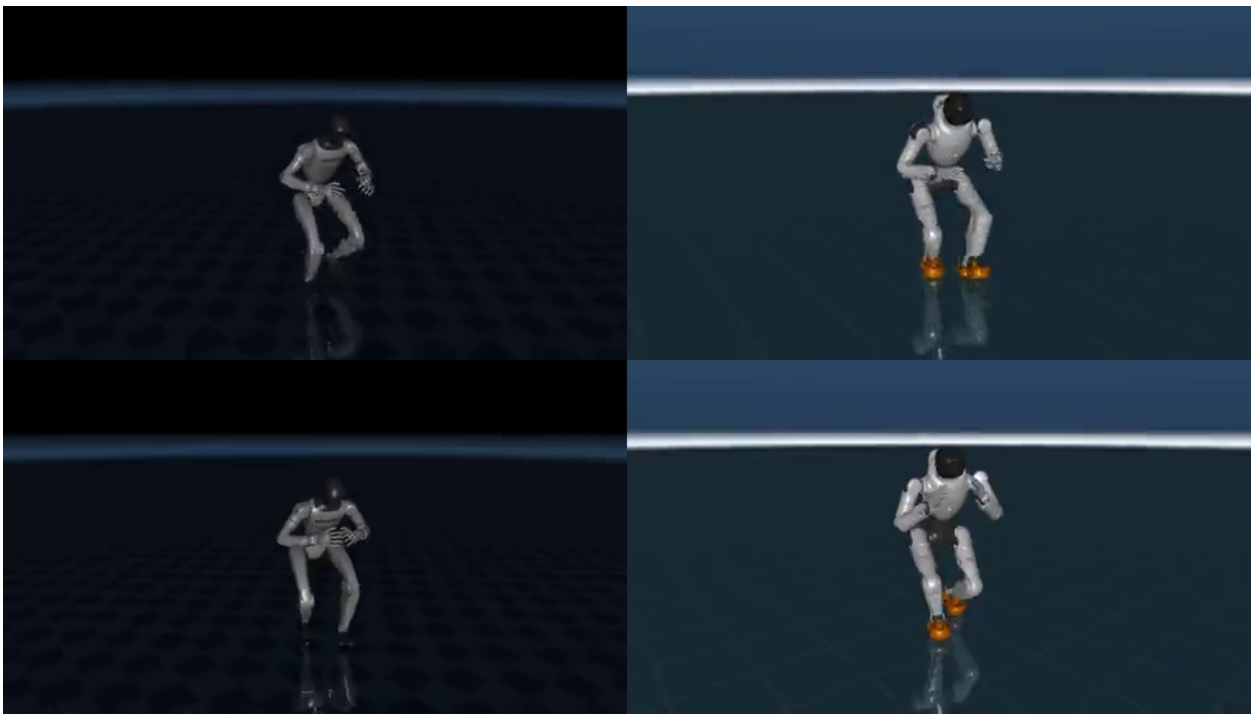

Figure 9: **F9c — Crouch walk (hip 0.55 m).** A capability neither native X2 planner had; inherited from the source core's crouch template for free. Depth saturates at the source corpus's floor (§9).

11 Limitations and Open Fronts

One embodiment pair, and a favorable one. Every result in this paper is one source-target pair, and the target's skeleton was deliberately built for weight reuse with the source's: identical 34-bone tree, matched representation widths, joint-for-joint naming correspondence (Section 5). We characterized the preconditions — calibration-exact arm affines, leg residuals absorbable by 83k parameters, matched intent semantics — but we have not tested where they break. An in-house stress pair exists (an embodiment missing waist and wrist joints, with sign flips and roughly half the source's range of motion) on which we expect the wrap to visibly strain; that experiment, which would map the method's boundary, has not been run. Until it is, the claim is "sibling embodiments," not more.

Simulation only. All closed-loop results are simulation with the shipped tracker checkpoint. The stack is deploy-shaped — the fused graph presents the deployed planner contract, the harness drives the real daemon, and latency is at parity — but hardware verification is future work, and the program's own history (Section 9's feel-test that passed a wrong-orientation build) counsels humility about any gate short of the robot.

One tracker. Closed-loop viability is defined here against a single tracker checkpoint. The float ablation shows the reference conventions that matter are *tracker-specific* — a retrained tracker could move the basin, invalidating the selected configuration (though not the selection protocol). We did not test configuration robustness across tracker versions beyond one incidental replication.

Known behavioral gaps, with owners. In-place turns slide (injected templates step in direct drive, not through the daemon path — the accepted inability); residual taps concentrate in tight turns (the operator's remaining complaint); crouch depth saturates at the corpus floor and its template awaits a calmer clip; first-stride initiation is improved but not polished; high-speed gait is out of envelope by attribution (core corpus + tracker coverage) and treated as a stress canary, not a target.

The remaining float. The reference still floats +1.2 cm; we know how to remove it (the height head fits to 1.3 mm) and that removing it costs ~3x recovery steps. The principled fix — a height target trained against tracker *trackability* rather than geometric ground truth (SONIC rollout scores as per-chunk regression weights) — is designed but unused. Likewise unused: closed-loop supervised training of the heads (CAT-K-style), which we expected to need and did not.

Mechanism gaps. The float's load-bearing-ness is established by ablation from both directions, but the mechanism is hypothesis-level (tracker training references carrying their own effective float; early-lift bias). The daemon-path turn-slide suspect (context filtering plus resampling flattening weight-shift steps) is likewise untested. And the recovery-step counter itself has known blind spots: it reads recorded weight shifts as extra steps at stop/start events and cannot see sub-threshold foot taps — the seam sweep's documented run-to-run variance means small differences between passing configurations ({1,2,1} versus {1,2,3}) should be read as parity, not ranking.

Selection cost. Closed-loop selection is the paper's method, and its price should be stated: every candidate costs rollouts (multi-seed sweeps plus tape replays), and the export-knob family search multiplies candidates. It was cheap here because the heads are tiny and simulation is fast; for a setting where rollouts are expensive, the anti-predictiveness of offline metrics we report is a genuinely inconvenient result.

12 Conclusion

We set out to buy a smooth slow walk for a demo and ended up with a result about model reuse. A generative motion planner trained to 2M steps on one humanoid drives a morphologically close humanoid at deployment quality with zero gradient steps on its weights: an analytic per-joint wrap achieves zero-shot transfer that beats both planners natively trained for the target robot, and 83k parameters of per-frame residual heads — selected in closed loop, tempered by export-time masking and strength scaling — close most of the remaining gap. The adaptation cost ~6 workstation-GPU hours against the ~10,000 H100-hours the source training schedule represents. For sibling embodiments — and we have characterized what "sibling" required here — frozen-planner reuse is practical today.

The transferable artifact, though, is the evaluation discipline, not the wrap. Three times in this program, offline metrics and closed-loop viability disagreed, and three times the offline metric was not just uninformative but anti-predictive: the natively trained planner that improved every offline number and lost in closed loop; the best-validation head architectures that destabilized or felled the tracker through deploy-time input distributions their training never showed them; the millimeter-exact height fit that removed a floating convention the tracker depends on. The working principles we extracted — select by rollout, never by reconstruction; audit every head input against what the pipeline (not the dataset) will feed it; treat the tracker as a projection operator and optimize basin margin, not exactness; keep absolute-geometry audits in the battery because stability gates can certify wrong references — are, we believe, worth more to a reader than the specific affine tables in Appendix A.

We reported the ledger in full: two closed-loop-falsified architectures (and a third rejected on validation) with their mechanisms, two reverted "fixes," two tape-parsing traps including one that passed a gate and a human feel-test, and a mode table that was wrong in a way that silently degraded every early result. The next experiments are named: the mismatched in-house embodiment where the wrap should strain, trackability-weighted height training for the last 1.2 cm of float, and the robot itself.

13 Acknowledgements

We thank **Nebius** for sponsoring the GPU credits that funded the cloud training fleet described in Section 4 — the SONIC tracker and both native planner-chain attempts were trained entirely on that infrastructure. We thank the MotionBricks authors for releasing the pretrained G1 planner checkpoints that made this study possible.

14 Authorship Note

This work was carried out as a human–AI collaboration. The first author set direction, constraints, and acceptance criteria, performed all robot and simulator evaluation driving, recorded the gait-initiation takes, and made every promotion decision. The AI co-author (Claude Fable 5, Anthropic, operating in Claude Code) implemented the wrap, training, and evaluation code, ran and analyzed the experiments under the first author's direction, maintained the experimental ledger, and drafted this manuscript, which the human authors reviewed. The human authors bear responsibility for the manuscript's claims. Venues that do not permit AI systems as authors should treat this note as the required disclosure, with the AI contribution moved to the acknowledgements.

15 Appendix A: Joint Alignment Tables

[Full per-joint affine table with fit statistics.]

16 Appendix B: Evaluation Harness

[Seam-sweep specification; the 16-frame daemon-faithful serving lesson; tape formats and parsing traps.]

17 Appendix C: Reproducibility

[Commits, seeds, run IDs, corpus manifest, exact commands.]

References

Aberman, Kfir, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. 2020. "Skeleton-Aware Networks for Deep Motion Retargeting." *ACM Transactions on Graphics* 39 (4). https://doi.org/10.1145/3386569.3392462.

Araújo, João Pedro, Yanjie Ze, Pei Xu, Jiajun Wu, and C. Karen Liu. 2025. "Retargeting Matters: General Motion Retargeting for Humanoid Motion Tracking." *arXiv Preprint arXiv:2510.02252*.

Harithas, Sudarshan, and Srinath Sridhar. 2025. "MotionGlot: A Multi-Embodied Motion Generation Model." In *IEEE International Conference on Robotics and Automation (ICRA)*.

He, Tairan, Zhengyi Luo, Wenli Xiao, Chong Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. 2024. "Learning Human-to-Humanoid Real-Time Whole-Body Teleoperation." In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Hu, Edward J., Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. "LoRA: Low-Rank Adaptation of Large Language Models." In *International Conference on Learning Representations (ICLR)*.

Jin, Ming, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, et al. 2024. "Time-LLM: Time Series Forecasting by Reprogramming Large Language Models." In *International Conference on Learning Representations (ICLR)*.

Kumar, Ananya, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. 2022. "Fine-Tuning Can Distort Pretrained Features and Underperform Out-of-Distribution." In *International Conference on Learning Representations (ICLR)*.

Lee, Sunmin, Taeho Kang, Jungnam Park, Jehee Lee, and Jungdam Won. 2023. "SAME: Skeleton-Agnostic Motion Embedding for Character Animation." In *SIGGRAPH Asia 2023 Conference Papers*. https://doi.org/10.1145/3610548.3618206.

Luo, Zhengyi, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris Kitani, and Weipeng Xu. 2024. "Universal Humanoid Motion Representations for Physics-Based Control." In *International Conference on Learning Representations (ICLR)*.

Silver, Tom, Kelsey Allen, Josh Tenenbaum, and Leslie Kaelbling. 2018. "Residual Policy Learning." *arXiv Preprint arXiv:1812.06298*.

Villegas, Ruben, Jimei Yang, Duygu Ceylan, and Honglak Lee. 2018. "Neural Kinematic Networks for Unsupervised Motion Retargetting." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 8639–48.

Wang, Tingwu, Olivier Dionne, Michael de Ruyter, David Minor, Davis Rempe, Kaifeng Zhao, Mathis Petrovich, et al. 2026. "MotionBricks: Scalable Real-Time Motions with Modular

Latent Generative Model and Smart Primitives." *ACM Transactions on Graphics* 45 (4). https://doi.org/10.1145/3811334.

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*.

Zhang, Jiaxu, Junwu Weng, Di Kang, Fang Zhao, Shaoli Huang, Xuefei Zhe, Linchao Bao, Ying Shan, Jue Wang, and Zhigang Tu. 2023. "Skinned Motion Retargeting with Residual Perception of Motion Semantics & Geometry." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, Lvmin, Anyi Rao, and Maneesh Agrawala. 2023. "Adding Conditional Control to Text-to-Image Diffusion Models." In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Zhang, Zhejun, Peter Karkus, Maximilian Igl, Wenhao Ding, Yuxiao Chen, Boris Ivanovic, and Marco Pavone. 2025. "Closed-Loop Supervised Fine-Tuning of Tokenized Traffic Models." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.