ACTPIN: CLOSED-LOOP SEMANTIC–SPATIAL CONTROL OF GENERATED MOTION

Anonymous authors
Paper under double-blind review

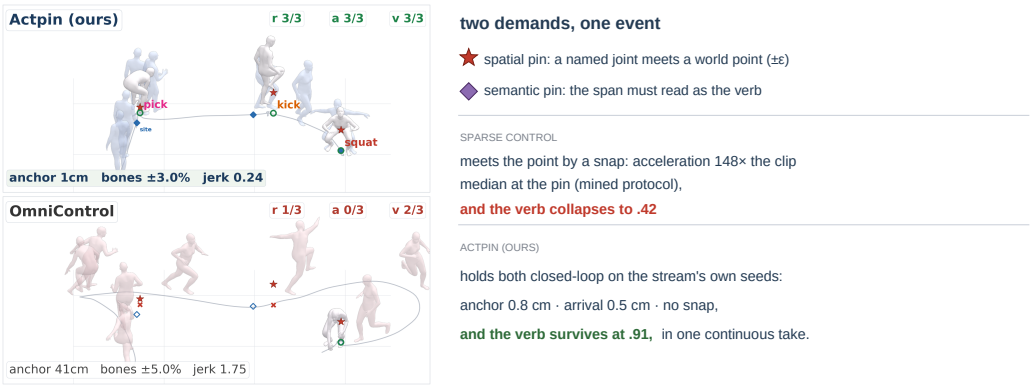

Figure 1: **Hold the verb where you pin it.** The authored input is one typed itinerary ("run, then *pick* ⋆; run, then *kick* ⋆; run, then *squat* ⋆"): per event a *spatial* pin (⋆, a named joint at a world point within ε) and a *semantic* one (the span must read as the verb); blue diamonds mark each leg's *site*, the arrival target the *pelvis* route must pass. Guidance *snaps* ($148\times$ on the mined protocol), token optimization places without performing (verb .51), imputation misses; we hold both closed-loop on a frozen stream's own seeds, in one take (Fig. 4). **This itinerary and every baseline tier are shown in motion in the supplementary demo video.**

ABSTRACT

Sparse spatial control of text-to-motion looks solved: drop an anchor, name the verb, and a guided or inpainted clip meets the point. We measure what that success is made of. When anchors are *mined from clips that already perform the verb*, a guided controller reads better than text alone ($.62 \rightarrow .92$), since the mined anchor narrows generation toward its source performance; yet the pin is met by a *snap*, the responsible joint accelerating at $148\times$ the clip median, invisible to retrieval. On freely *authored* anchors the verb collapses ($.24$–$.51$ across four families): every optimization these methods run serves space alone, so the point is bought by breaking the verb that no term defends. *Actpin* closes the loop: on a frozen motion-primitive stream whose *motion seeds* are the one differentiable input, it seeks per event the seed with legibility an objective measured on the rendered frames and the anchor and travel site both ε-constraints; across six coupling laws the strict read splits by whether the price *remembers* (we deploy the closed-form dynamic barrier). The deployed program holds every anchor at 0.7 cm and arrival at 0.5 cm, reads R@3 .90 / R@1 .71 averaged over three stream draws where the strongest baseline reads .51 / .28, and transfers to a held-out split.

1 INTRODUCTION

Imagine authoring motion the way a director walks a stage: standing in the scene, in XR glasses, you pin a mark in the air at the crate's corner and command "run here and *kick* at this mark, then cross over there and *kneel*." Nothing is captured or retargeted; the take must be *synthesized* from the marks alone. Each mark asks for two things at one event: a named joint must meet a world point at a commanded frame (the *anchor*), and the surrounding span must read as a named action (the *verb*).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Sparse spatial control of text-to-motion promises exactly this workflow, spanning guidance, imputation, densification, token-side optimization, and constraint-conditioned models (Karunratanakul et al., 2023; Xie et al., 2024; Cohan et al., 2024; Hwang et al., 2025; Pinyoanuntapong et al., 2025; Rempe et al., 2026), and on the standard evaluations the problem looks solved.

We measure what that success is made of, and it does not survive the audit (Sec. 2.2). Where the point is met at all, it is met by two artifacts: *the snap*, the body breaking at the commanded frame as the responsible joint teleports onto the point ($148\times$ the clip-median acceleration), and *the root shift*, the whole body sliding in with the pose frozen, so the action itself is not performed (40 cm body-frame error under sub-centimeter world error on the mined protocol). The standard evaluation sees neither, and worse: its anchors are mined from clips that perform the verb, so the anchor itself narrows generation toward that clip and *inflates* retrieval. On freely *authored* anchors, the marks of the scenario above, the cover is gone and the verb collapses (R@3 .24–.51): no real clip passes through an authored anchor, so there is nothing to inherit the verb from; the field *does* optimize, at the noisy sample or the token selection, but always for space alone, so when the two demands conflict the verb loses by construction.

Actpin puts both demands on *one* variable. We generate on a frozen autoregressive motion-primitive stream (Zhao et al., 2025) whose decoder emits only valid bodies and whose rollout composes to multi-event takes; its one differentiable input is the per-primitive *motion seed*, the latent noise that selects *which* plausible primitive the stream draws next. The seed does not touch coordinates, so it cannot snap a joint or slide a root, and unlike the noisy samples and token selections the field optimizes, it is a variable both demands can share. Actpin poses each event as one constrained program on its burst's seeds, *actpin seeding*: descend the verb, measured by a frozen motion–text encoder on the re-rendered frames, subject to the anchor, and to the arrival site, as hard ε-constraints, mid-stream, per event. The verb thereby becomes an *objective* scored on the output, where every baseline leaves it as a conditioning input.

Once both pins live on the same seeds, *how they couple* becomes the design object; the common practice, a fixed penalty weight on the constraint, is one point in that space, not its answer. The coupling is a price λ charged for violating the anchor while the verb descends, and the axis is whether the price *remembers*: an integral law accumulates it, a memoryless law recomputes it from the current state. Our setting decides it: the endgame lives on the ε-boundary, where an integral law meets each millimeter re-violation with its full accumulated price while a memoryless one releases inside the ball, freeing budget for the verb. We deploy the *dynamic barrier*, and a six-law study confirms the prediction: every law holds the anchor, only responsive prices bind the arrival ball, and the memoryless laws top both retrieval metrics.

On a tiered benchmark of authored itineraries (one to three actions in one native window, 240 reaches) the deployed program meets every anchor at 0.7 cm and every arrival at 0.5 cm with natural-range accelerations, and reads the verb at R@3 .90 / R@1 .71 (three-draw mean) against .51 for the strongest baseline, the read high across composition tiers while every baseline degrades. Our contributions: **(1)** a *quantified diagnosis* of sparse spatial control, measuring the breakage ($148\times$ accel, $2.2\times$ bone, 40 cm root shift) and the *protocol leak* that inflates mined-anchor retrieval ($.62 \rightarrow .92$) and hides the open loop; **(2)** *Actpin*: closed-loop two-pin control of a frozen motion-primitive stream on its motion seeds; **(3)** the first head-to-head *coupling design study* of constraint-price laws for generative motion; **(4)** the *Action Pinning Benchmark*: authored two-pin itineraries in three tiers, with validity axes and a held-out set.

2 The Action Pinning Problem

2.1 Problem definition

An *itinerary*, authored in bare absolute space with no host clip or scene mesh, is an ordered alternation of typed REACH and TRANSITION units. A REACH $e_i = (a_i,\, j_i,\, p_i^{\mathrm{tgt}},\, t_c^i)$ names an action a_i, its responsible joint j_i, a world anchor $p_i^{\mathrm{tgt}} \in \mathbb{R}^3$, and the *core frame* t_c^i at which the joint must meet it; a TRANSITION routes the body to the next site s_k^+ by its end frame. The output is one motion $X = (x_1, \ldots, x_T)$ on the 22-joint HumanML3D skeleton (the SMPL kinematic tree without its two hand joints; every method's native output is converted to these world joint positions for scoring, and meshes appear only in figures), the horizon set by the itinerary. With $\pi(\cdot)$ the pelvis ground

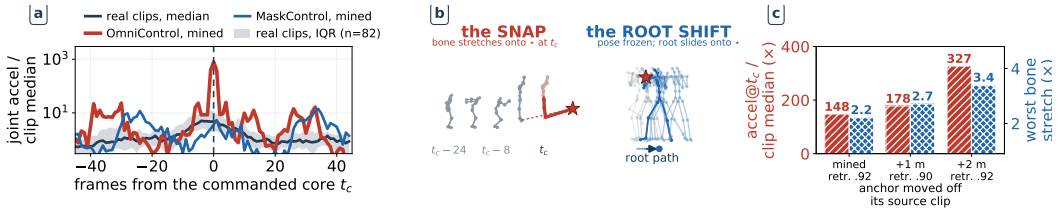

Figure 2: **Diagnosis on the home protocol (GT-mined anchors).** *(a)* Real performances accelerate ≈5× their clip median at the action's core (grey band); mined-anchor guidance spikes two orders above it at the commanded frame, token optimization stays in the band. *(b)* The two breakage types, each from the family that produces it: the *snap* (the joint teleports to ⋆) and the *root shift* (the body slides in with the pose frozen). *(c)* Both grow dose-wise as the mined anchor is shifted off its source clip; retrieval holds at .90+, blind. Protocol: Appendix B.

projection and $\mathcal{M}$ the physically valid motions, the task is

$$X^\star = \arg\min_{X \in \mathcal{M}} \sum_i \left\| x_{t_c^i}[j_i] - p_i^{\text{tgt}} \right\|_2 \; + \; \lambda \sum_k \left\| \pi(x_{t_k^e}) - s_k^+ \right\|_2, \tag{1}$$

subject to each span *reading as its type* under a motion–text encoder (TMR (Petrovich et al., 2023)). Every reach therefore *pins* twice at one event, a hard *spatial pin* and a *semantic pin*, and we call holding both the *action pinning* problem. The spatial sums alone are trivially zeroed by an implausible motion; the difficulty lives in $\mathcal{M}$ and in *semantic legibility* (details: Appendix A).

2.2 Diagnosis: why authored anchors break the field

No real clip passes through an authored anchor: there is no data prior at the point and no true trajectory to recover around it, and each family fails there through its own mechanism (audit: Table 15). *Guidance* edits coordinates toward the point and breaks the body, the *snap* and *root shift* of Fig. 2b, or misses outright when one window carries several actions (4.8 → 37 cm). *Imputation and densification* fill the masked frames as if the anchor were a keyframe of some real trajectory; none exists, so the completion invents one and misses it (70–99 cm). *Token optimization* places cleanly (1.5 cm), but its re-selection objective has a spatial term and no verb term, so it converges on whatever plausible motion passes the point (strict read .28 R@1 on authored anchors, Table 1). The root cause is not a lack of optimization: every such objective is *spatial*, on a variable the semantic condition does not touch (Table 3, appendix), so nothing arbitrates the conflict (R@3 .24–.51).

Why has this not been visible? The standard protocol validates sparse control with *GT-mined* targets: spatial signals read off dataset clips that already perform the verb (Karunratanakul et al., 2023; Xie et al., 2024; Cohan et al., 2024; Pinyoanuntapong et al., 2025). Fig. 2 re-runs that protocol (our replication; Appendix B) and finds it certifies the wrong thing: a mined anchor is a fragment of a real performance, so conditioning on it narrows generation toward that performance and *inflates* retrieval over text alone (.62 → .92; MaskControl reads .96 mined vs. .51 authored), while the breakage grows dose-wise as the anchor leaves its source clip and retrieval stays .90+, blind (Fig. 2c). The diagnosis fixes what a solution must provide: the verb as an *objective* on the rendered output, held jointly with a hard anchor, on a channel that cannot break the body (Sec. 3).

2.3 Related work

Fixed-window sparse spatial control. GMD (Karunratanakul et al., 2023), OmniControl (Xie et al., 2024) and CondMDI (Cohan et al., 2024) inject world targets by guidance or imputation, SFControl (Hwang et al., 2025) densifies sparse keyjoints, and TLControl/MaskControl (Wan et al., 2024; Pinyoanuntapong et al., 2025) bring trajectory control to token generation. All are bidirectional models over one ≈196-frame window that pull the joint to the target *inside* the window; as measured above, the met anchor is bought with breakage. Scene-bound synthesis (LINGO (Jiang et al., 2024), SCENIC (Zhang et al., 2024), Story-to-Motion (Qing et al., 2023)) is closest in *task* but generates inside voxelized or distance-field scene encodings, or retrieves and stitches library clips; we generate from scratch, from typed reach and transition events alone (capability landscape: Table 2, appendix).

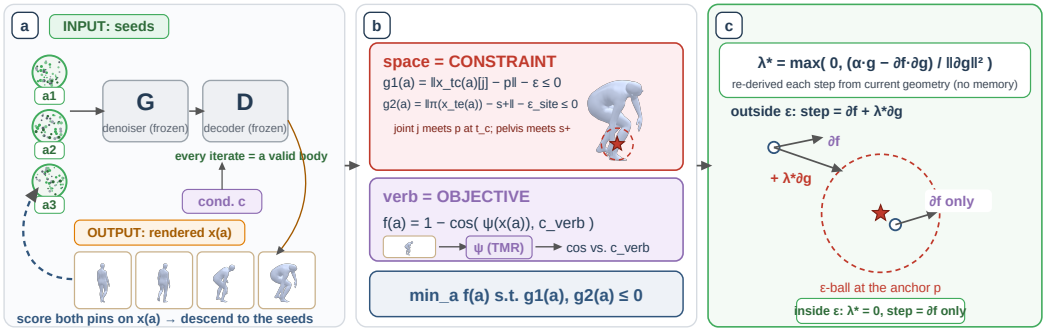

Figure 3: **Actpin: one constrained program on the stream's motion seeds.** *(a)* The frozen chain renders seeds into frames (the strip shows four sampled frames of one burst); both pins are scored on the frames and descend back to the seeds. *(b)* Space enters as hard ε-constraints, the anchor and the arrival site, and the verb as an objective read on the rendered output, one program on the seeds. *(c)* The deployed barrier prices each violated constraint from current geometry, one multiplier per constraint, and *releases* inside the ε-ball (Eq. 8; studied in Sec. 4.4).

Streaming and autoregressive generation. DART (Zhao et al., 2025), MotionStreamer (Xiao et al., 2025), PRIMAL (Zhang et al., 2025) and COMET (Lee et al., 2026) remove the horizon limit, yet each carries at most one of the two channels the task needs (text *or* per-joint targets, not a world anchor *and* a named action together); CLoSD (Tevet et al., 2025) and UniPhys (Wu et al., 2025) add a physics loop our kinematic setting does not assume. We adopt DART's rollout and shape-maintenance regime but replace its conditioning, driving locomotion and actions from one motion–text-aligned embedding space.

Noise optimization and constraint coupling. Test-time optimization itself is everywhere above: guidance descends the noisy sample, MaskControl its token logits; the distinction is not optimizing versus not optimizing, but *which objective on which variable*. Optimizing a diffusion model's initial noise is likewise established offline (DNO (Karunratanakul et al., 2024); ReNO/InitNO/DOODL (Eyring et al., 2024; Guo et al., 2024; Wallace et al., 2023); best-of-N (Ma et al., 2025)) on a *finished* clip, and DART itself steers its stream's seeds toward spatial goals; Actpin seeding (Sec. 3.2) puts a different program on that lever: a semantic objective held against hard spatial ε-constraints, per event, mid-stream. Once both pins live on those seeds, how they *couple* is a constrained-optimization question with a long menu of price laws (quadratic penalty (Courant, 1943), augmented Lagrangian (Hestenes, 1969), primal–dual (Arrow et al., 1958), PID-Lagrangian (Stooke et al., 2020), dynamic barrier (Gong et al., 2021)) that generative-motion work has not compared: everywhere above, the coupling is a weight fixed once, including constraint-conditioned models like Kimodo (Rempe et al., 2026). Holding the verb closed-loop under a hard anchor, and treating the coupling as an object of study (derived in Sec. 3.3, tested in Sec. 4.4), is what we add.

3 Method

Figure 3 shows the whole method: a frozen motion-primitive substrate (Sec. 3.1); *actpin seeding*, the seed that satisfies both pins and the optimization that finds it (Sec. 3.2); and the coupling that prices the constraint (Sec. 3.3).

3.1 Preliminary: A frozen motion-primitive stream

We generate on DART (Zhao et al., 2025), a frozen autoregressive motion-primitive diffusion model; its anatomy is not our contribution. A denoiser G draws a primitive latent z from a *motion seed* a (the primitive's initial noise: under a fixed history and condition it selects *which* plausible primitive is drawn), and a decoder $\mathcal{D}$ reconstructs a valid SMPL body from it, so every iterate has rigid bones and plausible contact. Our one adaptation replaces the CLIP-text slot with a learned head W reading per-event *action embeddings* c (Sec. 3.4). The itinerary is *one* stream,

$$z_{k+1} = G\big(a_k,\, h_k,\, W(c_k)\big), \qquad x_{k+1} = \mathcal{D}(z_{k+1}, h_k), \tag{2}$$

where k indexes primitives, h_k is the canonicalized history so far, c_k the current event's embedding, and x_{k+1} the decoded frames; travel seeds are written by a goal-reaching policy driving the same denoiser, so momentum carries across every boundary (details: Appendix G). The embeddings c_k come from a fixed per-class bank for reaches and, for transitions, from a context predictor, the *Transition Bond* (both are input machinery: Sec. 3.4). Given embeddings, this stream runs open-loop and hits nothing: the anchors are still unmet.

3.2 ACTPIN SEEDING

3.2.1 THE TWO PINS AS ONE PROGRAM

The anchor is *not* navigation, so the travel policy cannot supply it, and the shape-guaranteeing decoder forbids meeting it by editing frames; the event must be placed by *choosing the seed*. It is also where the two demands can finally share one program: the field optimizes the noisy sample or the token selection, stages the semantic condition cannot reach, whereas every seed update re-renders the whole burst both pins are scored on. We make both demands of Sec. 2.1 measurable on the rendered burst $x(a)$: the *spatial pin* is the hard constraint

$$g(a) = \left\| x_{t_c}(a)[j] - p^{\mathrm{tgt}} \right\| - \varepsilon \;\le\; 0, \tag{3}$$

and the *semantic pin* scores the burst in the reader's own space,

$$f(a) = 1 - \cos\big(\psi(x_{1:T_b}(a)),\, c_v\big), \tag{4}$$

with ψ the frozen TMR motion encoder (Petrovich et al., 2023) applied to the burst's *entire* rendered frame sequence $x_{1:T_b}(a)$ (the reader scores spans, not a single frame) and c_v the verb's prototype, the mean TMR embedding of real clips captioned with that verb (prototypes from the conditioning bank optimize their own cosine yet leave retrieval unmoved; Table 14). An *actpin seed* is a motion seed whose rendered burst satisfies both pins at once, and finding one is a constrained program on the seeds:

$$a^\star = \arg\min_a \; f(a) \qquad \text{s.t.} \qquad g(a) \le 0. \tag{5}$$

In the deployed program g carries *two* spatial constraints of the same form, the anchor and the travel leg's arrival site, $g_2(a) = \|\pi(x_{t_e}(a)) - s^+\| - \varepsilon_{\text{site}} \le 0$ (the pelvis at the hand-over, the notation of Eq. 1), each with its own multiplier (Appendix H). The two pins deliberately bind at different granularities: the spatial pin constrains the single commanded frame, which is what the task pins (Eq. 1), while the semantic pin reads the *whole* span, and the auxiliary terms below shape the frames around t_c, so the core's neighborhood is governed though only the core is constrained (re-timing the action's own impact toward t_c is analyzed in Appendix J).

3.2.2 FINDING THE SEED: THE IN-STREAM DESCENT

Actpin seeding solves this program *in place*, treating the frozen denoiser-plus-decoder as one differentiable renderer from seeds to frames. At each reach it starts from the seeds the stream drew for that burst (a^0) and, for 80 steps, re-renders the burst, scores both pins on the frames, and backpropagates through the frozen chain to the seeds; the optimized $a^\star$ replace the drawn seeds and the rollout continues downstream unchanged. The descent target is a working loss

$$\mathcal{L}(a) = f(a) + \lambda(t)\, g(a) + \mathcal{L}_{\text{aux}}\big(x(a)\big) + \lambda_{\text{tr}} \big\| a - a^0 \big\|_2^2, \tag{6}$$

where $\lambda(t)$ is the coupling of Sec. 3.3 and the last term is a *trust region* around the stream's own draw a^0, keeping the optimized seed a plausible draw. We do not tune λ_{tr} as an explicit weight: the trust region is implemented as the descent's step budget itself, a per-seed L_2-normalized gradient under a linearly decaying learning rate for the 80 steps, which bounds $\|a - a^0\|$ by construction (Appendix H). The auxiliary terms close the three escapes unconstrained descent would take out of the stream's support:

$$\mathcal{L}_{\text{aux}}(x) = \lambda_v \big(\|\dot p_{\text{in}}\| - \|\dot p_0\| \big)^2 + \lambda_f \sum_{f \in \text{feet}} \text{relu}(-x[f]_z) + \lambda_p \,\text{relu}(z_0 - \delta - x[0]_z), \tag{7}$$

a velocity-continuity term (deceleration is spent inside the reach), a floor term, and a verb-gated pelvis-stay term (a low target should be reached by posture, not by sinking); settings and cost in Appendix H. This is not guidance: guidance perturbs *coordinates* inside a fixed window and can

bend the body, while seeding re-chooses the *seed* and re-renders through the frozen decoder, so every iterate is a valid body, mid-stream on a rollout.

3.3 COUPLING: PRICING THE SPATIAL PIN

We do *not* add the pins: a weighted sum lets either pin buy the other, and projecting the semantic gradient off the pin gradient does not hold a level set. What remains is a price $\lambda(t)$ charged for violating the constraint. The update $a \leftarrow a - \gamma(\nabla_a f + \lambda \nabla_a g)$ is shared, and the *law* that sets λ is the design choice. The axis that organizes the classical menu is *memory*: an *integral* law accumulates its price over the violation history ($\lambda \leftarrow \lambda + \mu g$, summed over steps: primal–dual, PID, ALM), while a *memoryless* law recomputes the price from the current state alone. Section 3.2 fixes three requirements that decide between them. *(R1) The landscape moves*: every update re-renders the burst, so a remembered price prices a landscape that no longer exists. *(R2) The violation is one-sided*: the violation $\max(0, g)$ cannot go negative, so an integrated price only grows while the anchor is unmet and has no mechanism to come back down after it is met. *(R3) The budget is short*: roughly half of the 80 descent steps happen *after* the anchor is first captured, and a price that cannot release spends that half fighting the verb instead of funding it. Table 12 (appendix) lays out the menu. We deploy the memoryless law whose one rate is *geometric* rather than scheduled, the dynamic barrier:

$$\lambda^\star = \max\Big(0,\ \tfrac{\alpha\, g(a) - \langle \nabla_a f,\ \nabla_a g \rangle}{\|\nabla_a g\|^2}\Big). \tag{8}$$

In two-pin terms, Eq. 8 compares what the constraint needs (close g at rate α) with what the semantic step already provides ($\langle \nabla f, \nabla g \rangle$): a verb step that already moves the joint toward the anchor is not charged, one that pulls away is charged exactly enough to restore the closing rate; with g the one-sided violation of R2, $\lambda^\star{=}0$ exactly inside the ε-ball and the whole budget goes to the verb. The one hyperparameter α is a return rate to the feasible set, geometric and unitless, which is why the barrier tunes flatter than a loss-scale-coupled schedule (tested head-to-head in Sec. 4.4).

3.4 CONDITIONING: THE ACTION EMBEDDING (SUPPORTING DETAIL)

The rollout consumes one condition per primitive, the c_k of Eq. 2; this subsection fixes that *input* (control lives in the seeds, and ablating this block is measurably free: Table 16, appendix). Each event is conditioned by an *action embedding* on the TMR-aligned unit sphere $\mathbb{S}^{255} \subset \mathbb{R}^{256}$ (a 256-d unit vector), the presence- and length-weighted mean of its per-track segment embeddings, so one limb can act while others carry a gait; a fixed bank, built offline once, holds one embedding per action class and locomotion type. A reach's embedding is fixed, and a transition's is *predicted* from its context by the *Transition Bond*,

$$\hat{g} = F_\phi\big(c_{\text{prev}},\, c_{\text{next}},\, c_\ell,\, r\big) \in \mathbb{S}^{255}, \qquad r = (d,\ \Delta t,\ v,\ |\Delta\theta|), \tag{9}$$

where $c_{\text{prev}}, c_{\text{next}}$ are the flanking reaches' action embeddings, c_ℓ the commanded gait's embedding, and r the leg's route geometry (site displacement d, span duration Δt, implied speed v, heading change $|\Delta\theta|$), so that the approach to a careful squat is measured and to a dynamic kick brisk. The prediction $\hat{g}$ enters the stream exactly where a reach carries its class embedding: it is the c_k of Eq. 2 on every transition primitive. The bond sets the travel's *manner*, not the reach's action; the model and its evaluation are in Appendix D (the bond predictor).

4 EXPERIMENTS

4.1 SETUP

Generation runs on the frozen base of DART (Zhao et al., 2025): the shape-guaranteeing decoder is not trained at all; its denoiser and the head W are finetuned once to read the embedding bank in place of text (23.5M, $\sim$1.7 h); the controller is a PPO policy (3.8M); nothing is retrained per itinerary. We compare against OmniControl, MaskControl, CondMDI, SFControl, and Kimodo, all with authors' weights, each run in its native single window. The table's DART† row is the same frozen substrate with our conditioning and seeding off: the closest reading of DART on this task, and our open-loop floor. The diagnosis used GT-mined anchors; everything here runs *authored* anchors, where nothing leaks the verb.

Table 1: **Main comparison on the Action Pinning Benchmark** (40 itineraries per tier of 1/2/3 actions, 240 reaches, every itinerary inside one native window; per-tier breakdown in Table 9, appendix; metric definitions in Sec. 4.2). DART† is the frozen substrate under our conditioning with seeding off. **Bold** = best among the placers (≤2 cm anchor); Accel@t_c, Jerk and PJ@join are calibrated axes (GT 5×/2.5/.22, closest to GT is best, so no bold). Accel@t_c is read where the anchor is met; on rows missing by ≥70 cm too few reaches qualify (–). Ours row = mean over three stream draws (seeds 42–44; arrival 0.5 cm on every draw); per-draw and held-out rows in Table 14.

Method	Anchor↓	Arrival↓	Bone %↓	Accel@t_c (→GT 5)	Jerk (→GT 2.5)	PJ@join (→GT .22)	R@1↑	R@3↑
Sparse spatial controllers (each in its native single window)								
OmniControl (Xie et al., 2024)	24.8	5.7	3.4	36.1	43.6	1.64	.158	.421
MaskControl (Pinyoanuntapong et al., 2025)	1.5	**0.5**	4.9	2.2	2.8	.12	.283	.508
CondMDI (Cohan et al., 2024)	72.8	25.3	1.5	–	4.7	.21	.125	.321
SFControl (Hwang et al., 2025)	99.2	45.6	4.8	–	11.5	.33	.167	.383
Kimodo (Rempe et al., 2026)	70.1	4.0	0.4	–	19.0	11.6	.062	.242
Stream-native, ours ablated (the closing-the-loop delta)								
DART† (Zhao et al., 2025) (seeding off)	70.6	9.0	1.8	–	1.7	.12	.450	.704
anchor pin only (open-loop verb)	**0.55**	6.6	**2.2**	2.8	3.1	.16	.463	.800
two-pin, closed loop (− arrival ext.)	1.34	6.6	2.4	2.6	3.0	.15	**.750**	**.963**
Ours (full program, deployed)	0.73	**0.5**	2.5	2.5	3.2	.14	.710	.903

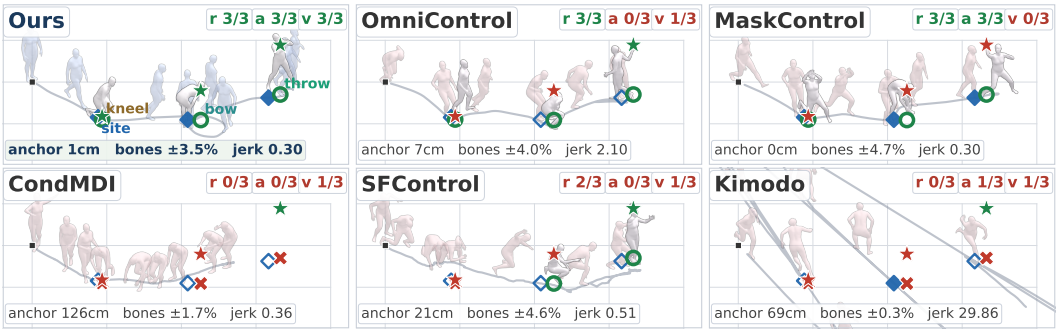

Figure 4: **Qualitative comparison on one tier-3 APB itinerary** (kneel → bow → throw, 189 f). Per event: circle = the *spatial* pin (green = anchor met, red X = missed); star at the authored anchor = the *semantic* pin (green = the span reads as its action, TMR top-3); blue diamond = each leg's *site*, the arrival target, filled when the *pelvis* route passes within 1 cm (2× the deployed $\varepsilon_{\text{site}}$, absorbing frame sampling). Badges per panel: **r** = anchors reached, **a** = arrivals met (within 1 cm), **v** = verbs read (TMR top-3). Placement and legibility separate: MaskControl reaches 3/3 anchors with 0/3 verbs, ours holds both.

4.2 The Action Pinning Benchmark

The *Action Pinning Benchmark* (APB) consists of 120 authored itineraries (built from scratch, no host clip or ground-truth motion) in three *difficulty tiers* of 40: one, two, or three actions, 240 reaches in all, each preceded by a forced locomotion segment. Every itinerary fits a single native window (≤ 190 f), so every method produces every reach and difficulty is composition, not coverage. Table 1 reports, per reach or join: *anchor* is the responsible-joint-to-anchor distance at the core frame (cm), and *arrival* the closest pelvis-to-site pass through the hand-over (held by our program as a second ε-constraint; a hard pelvis keyframe for the baselines). *Bone* is the per-bone length variation over time (%), and *jerk* the whole-body third difference. *Accel@t_c* is the responsible joint's acceleration at the commanded frame over the clip median, the snap axis, read where the anchor is met. *PJ@join* is FlowMDM's Peak Jerk (Barquero et al., 2024) on the travel→reach boundary (a finer pelvis-speed diagnostic in Appendix I). *R@1/R@3* follow TMR's retrieval protocol (Petrovich et al., 2023): each realized reach span retrieves its class caption among the 13 (chance R@3 ≈.23). Construction, the held-out split, the timing contract, and redraw stability are in Appendix F and Table 14.

4.3 Main comparison

Table 1 places the whole field on the tiered task, and Fig. 4 renders one itinerary side by side. The baselines place the joint or read the verb, not both. MaskControl is the strongest placer (1.5 cm)

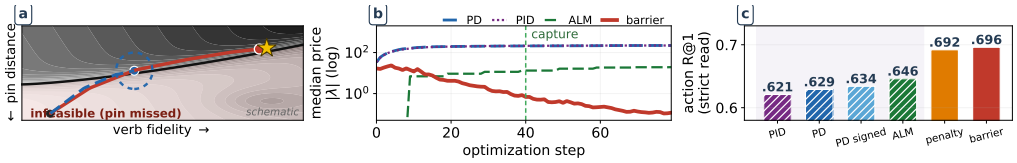

Figure 5: **The coupling answer at a glance.** *(a)* The verb optimum sits at the feasible ball's edge. *(b)* Median price $|\lambda|$ per step: every law captures at $\sim$step 40, then the barrier *releases* while integral prices hold their level. *(c)* The *strict* read splits exactly by memory class (hatched = integral); anchors do not separate, the arrival ball does (Table 13).

yet reads .508, with the loosest skeleton of the placers (bone 4.9% vs. our 2.5) and a comparable seat on the real-motion manifold (span-FID, Appendix J). OmniControl meets what it meets by snapping (36× the clip median), and CondMDI, SFControl and Kimodo miss placement outright (70–99 cm). On our own substrate the open-loop condition reproduces the collapse in softened form: anchor enforced, verb left as a condition, the read is .800 open-loop against .963 closed (.463 vs. .750 R@1). The difficulty axis isolates composition (Table 9, appendix): from one action to three, every baseline degrades; the strongest placer's verb collapses (.725 → .433), guidance loses even its placement (4.8 → 37 cm), while ours holds a sub-centimetre anchor at every tier (0.66/0.74/0.74, three-draw means) with the read high on tiers 1–2 (R@3 .94 on both) and lower on tier 3 (.86; .858–.875 over draws).

Averaged over three stream draws the deployed program lands at 0.73 cm anchor and 0.5 cm arrival, matching the strongest placer's arrival exactly, with no snap (2.5×) and R@3 .903 / R@1 .710 (per-draw R@3 .896–.913, .921 on the held-out split; arrival 0.5 cm on every draw), which is +.40 R@3 over the best baseline. On the standard transition metric the stream does not tear (PJ@join .12–.16 vs. the GT entry .22), but jerk cannot see a *rhythm* break. A stricter pelvis-speed diagnostic calibrated on real entries (Appendix I) shows the bare two-pin decelerating *at* the hand-over (8.0× vs. the GT band 1.2–3.9); constraining the site to 0.5 cm arrival buys back most of it (4.2–4.4×, just above the band; a light site penalty sits inside at 3.8×, Table 14), and the same diagnostic reads the anchor-only arm's *missing* entry (0.1×). The program transfers to a fully re-authored seed-disjoint split at R@3 .921, and on the diagnosis's GT-mined events it meets the same anchors an order of magnitude smoother (8–11× vs. 148×) at R@3 1.00 converged (Table 15).

4.4 The coupling design study

Section 3.3 predicted that the *read* should split the laws by whether the price remembers, while the constraints should separate nothing; Table 13 (appendix) is the test, each law running the deployed two-constraint program (anchor and site, one multiplier per constraint, n=1440 reaches). The anchor indeed separates nothing (means 0.52–0.81 cm, zero divergent runs), but the *second* constraint does: the slow-integrating laws never bind the arrival ball within the 80-step budget (primal–dual, PID and the signed variant sit at 1.4–2.2 cm), while the memoryless pair and the safeguarded ALM hold it at 0.5 cm. On the read the memoryless pair tops both metrics (barrier .913/.696, penalty .929/.692, their gap inside draw noise) and every integral law sits below on both (.887–.892 R@3, .621–.646 R@1; Fig. 5c); the barrier wins 59–67% of decided per-reach comparisons against the integral laws and splits evenly with the penalty. The same strict-read split held across draws at the pre-extension layer (.750–.754 vs. .717–.721).

The mechanism is the price after capture (Fig. 5a): the verb optimum lies at the feasible ball's edge, so after capture (median step $\sim$40, Fig. 5b) the endgame runs on the boundary under constant millimeter re-violations, which an integral law meets with its full accumulated λ and a memoryless law only in proportion. The moving-landscape premise (R1) is measurable: along the descent the verb gradient's direction is stable between steps (median cosine .97) while the constraint gradient's is not (.92, a quarter of steps below .5), so an accumulated price keeps paying for a geometry that no longer exists. The barrier tunes flattest (α tenfold: $\sim$.02).

4.5 Ablation: actpin seeding, taken apart

Figures 7 and 7 (appendix) apportion the result; the full tables are Tables 16 and 14, rendered in Appendix N. Each pin carries its own axis. With seeding off the anchors miss by 71 cm. The spatial

pin alone places them at half a centimetre, but the read stays at .800: placement does not buy the verb. Adding the semantic pin lifts the read to .963 at a still-held anchor, so the second pin is worth +.16 R@3 and +.29 R@1. The semantic target must also live in the reader's space, since class-bank prototypes cap the read near .80 where real-clip TMR prototypes unlock it. The coupling must be constrained rather than summed: a weighted sum trades the anchor away by centimetres, while under the constrained laws the anchor tracks ε exactly across the whole frontier (Table 14). Finally, the two demands must share the descent. Holding everything fixed except the step at which the semantic objective joins, the read climbs monotonically from the anchor-only arm to the full two-pin arm, and joining only for the last quarter buys almost nothing, so the verb is not a polish applied after placement (Fig. 6, appendix). Validity gaming and the one-term extensions are quantified in Appendix L.

5 Discussion

Placement machinery is not what separates the methods (MaskControl places as precisely as we do); what separates them is whether anything defends the verb while the point is met, and composition makes the split visible: from one action to three the baselines lose their verb while the stream's per-event program keeps the read flat. The missing mechanism is not more optimization, which the field already runs, but the per-event loop on a *shared* variable, where the verb can push back the moment space pushes; and the verb must sit on *both* sides of that loop, in the prior (conditioning the whole event) and in the objective; restricting either side costs the read at unchanged anchors (Appendix H).

Limitations. Semantic locomotion covers the three commanded gaits, and the action bank is closed at its 13 classes (caption-aligned: an open bank is a data question, not an architectural one). The residual misses have one shape, two-handed or whole-posture verbs (clap, reach-up, bow) whose anchored wrist arrives while the rest of the body's evidence does not follow. The spatial pin constrains the core frame, not the action's own impact timing; pricing the stroke's peak against the core frame is the natural next term (Appendix J). Arrival enters the baselines as a literal keyframe while our program holds it as a second ε-constraint (arrival 0.5 cm for $-.029$ R@3 against the site-free variant; a light penalty instead keeps R@3 .938 at 3.0 cm arrival, Table 14), and a minority of travel legs still realize a near-goal *orbit* (Appendix G). Route *properties* are uncontrolled beyond the commanded gait: shortest-path or clearance-aware travel, entry-speed conditioning that matches each action's own approach profile, and grounded multi-body itineraries (the physics-loop line (Tevet et al., 2025; Wu et al., 2025)) are future work; and since every reach is its own re-solvable program, an *editing* pipeline that re-pins one event mainly needs cached per-burst histories.

6 Conclusion

Sparse spatial control looks solved; the success is a snap, a root shift, and an open loop that collapses on the authored anchors real authoring needs. Actpin closes the loop inside generation, the anchor a hard ε-constraint and the verb an objective measured on the output, both descended on a frozen stream's own motion seeds through a memoryless coupling; nothing is retrained, and each reach resolves the program in place. The deployed program holds every anchor at 0.7 cm and every arrival at 0.5 cm, both pins explicit ε-constraints, lifts R@3 .80 $\rightarrow$ .90 and R@1 .46 $\rightarrow$ .71 over its open-loop condition, transferring to a held-out split; the coupling study adds a design result of its own, memory in the price separating the strict read. The diagnosis, benchmark, and audit metrics are as much the contribution as the method: they measure the open loop mined-anchor protocols have hidden.

Ethics statement

All experiments use publicly released datasets (HumanML3D, built on AMASS) and publicly released model weights under their licenses; no human subjects, personal data, or demographic attributes are involved. Controllable motion synthesis could in principle serve deceptive animation; our contribution is a measurement protocol and a control method for existing generators, which makes failure modes visible.

AI USE STATEMENT

The research ideas, problem formulation, method, and experimental design are the authors'. We used generative AI tools to assist in carrying out the work: executing and monitoring author-specified experiment runs, drafting analysis code, building visualization tooling, literature search, and editing of author-written text. The authors reviewed and verified all AI-assisted code, numbers, and artifacts, and take full responsibility for the content of this work.

REPRODUCIBILITY STATEMENT

The problem, benchmark, method, and coupling study are specified in Secs. 2–4, with the substrate, deployed settings, metric definitions, per-method inputs, and the full benchmark listing in the appendix (Appendices A–O). All runs use authors' released baseline weights and a single frozen generator; the benchmark generator, specs, evaluation code, and our steering code will be released.

REFERENCES

Kenneth J. Arrow, Leonid Hurwicz, and Hirofumi Uzawa. *Studies in Linear and Non-Linear Programming*. Stanford University Press, 1958.

German Barquero, Sergio Escalera, and Cristina Palmero. Seamless human motion composition with blended positional encodings. In *CVPR*, 2024. arXiv:2402.15509.

Setareh Cohan, Guy Tevet, Daniele Reda, Xue Bin Peng, and Michiel van de Panne. Flexible motion in-betweening with diffusion models. In *SIGGRAPH*, 2024.

Richard Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin of the American Mathematical Society*, 49(1):1–23, 1943.

Luca Eyring, Shyamgopal Karthik, Karsten Roth, Alexey Dosovitskiy, and Zeynep Akata. ReNO: Enhancing one-step text-to-image models through reward-based noise optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2406.04312.

Chengyue Gong, Xingchao Liu, and Qiang Liu. Automatic and harmless regularization with constrained and lexicographic optimization: A dynamic barrier approach. In *NeurIPS*, 2021.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3D human motions from text. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Xiefan Guo, Jinlin Liu, Miaomiao Cui, Jiankai Li, Hongyu Yang, and Di Huang. InitNO: Boosting text-to-image diffusion models via initial noise optimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. arXiv:2404.04650.

Magnus R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, 1969.

Inwoo Hwang, Jinseok Bae, Donggeun Lim, and Young Min Kim. Motion synthesis with sparse and flexible keyjoint control. In *International Conference on Computer Vision (ICCV)*, 2025. arXiv:2503.15557.

Nan Jiang, Zimo He, Zi Wang, Hongjie Li, Yixin Chen, Siyuan Huang, and Yixin Zhu. Autonomous character-scene interaction synthesis from text instruction. In *SIGGRAPH Asia*, 2024. arXiv:2410.03187.

Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. Guided motion diffusion for controllable human motion synthesis. In *IEEE International Conference on Computer Vision (ICCV)*, 2023.

Korrawe Karunratanakul, Konpat Preechakul, Emre Aksan, Thabo Beeler, Supasorn Suwajanakorn, and Siyu Tang. Optimizing diffusion noise can serve as universal motion priors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. arXiv:2312.11994.

Eunjong Lee, Eunhee Kim, Sanghoon Hong, Eunho Jung, and Jihoon Kim. Controllable long-term motion generation with extended joint targets. In *WACV*, 2026.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025.

Mathis Petrovich, Michael J. Black, and Gül Varol. TMR: Text-to-motion retrieval using contrastive 3D human motion synthesis. In *International Conference on Computer Vision (ICCV)*, 2023.

Ekkasit Pinyoanuntapong, Muhammad Usama Saleem, Korrawe Karunratanakul, Pu Wang, Hongfei Xue, Chen Chen, Chuan Guo, Junli Cao, Jian Ren, and Sergey Tulyakov. MaskControl: Spatio-temporal control for masked motion synthesis. In *International Conference on Computer Vision (ICCV)*, 2025. arXiv:2410.10780.

Zhongfei Qing, Zhongang Cai, Zhitao Yang, and Lei Yang. Story-to-motion: Synthesizing infinite and controllable character animation from long text. *SIGGRAPH Asia Technical Communications*, 2023. arXiv:2311.07446.

Davis Rempe, Mathis Petrovich, Ye Yuan, Haotian Zhang, Xue Bin Peng, Yifeng Jiang, Tingwu Wang, Umar Iqbal, et al. Kimodo: Scaling controllable human motion generation. *arXiv preprint*, 2026. arXiv:2603.15546.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by PID Lagrangian methods. In *ICML*, 2020.

Guy Tevet, Sigal Raab, Setareh Cohan, Daniele Reda, Zhengyi Luo, Xue Bin Peng, Amit H. Bermano, and Michiel van de Panne. CLoSD: Closing the loop between simulation and diffusion for multi-task character control. In *ICLR*, 2025. arXiv:2410.03441.

Bram Wallace, Akash Gokul, Stefano Ermon, and Nikhil Naik. End-to-end diffusion latent optimization improves classifier guidance. In *International Conference on Computer Vision (ICCV)*, 2023. arXiv:2303.13703.

Weilin Wan, Zhiyang Dou, Taku Komura, Wenping Wang, Dinesh Jayaraman, and Lingjie Liu. TLControl: Trajectory and language control for human motion synthesis. In *European Conference on Computer Vision (ECCV)*, 2024.

Yan Wu, Korrawe Karunratanakul, Zhengyi Luo, and Siyu Tang. UniPhys: Unified planner and controller with diffusion for flexible physics-based character control. In *ICCV*, 2025.

Lixing Xiao, Shunlin Lu, Huaijin Pi, Ke Fan, Liang Pan, Yueer Zhou, Ziyong Feng, Xiaowei Zhou, Sida Peng, and Jingbo Wang. MotionStreamer: Streaming motion generation via diffusion-based autoregressive model in causal latent space. In *ICCV*, 2025. arXiv:2503.15451.

Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. OmniControl: Control any joint at any time for human motion generation. *International Conference on Learning Representations (ICLR)*, 2024.

Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. PhysDiff: Physics-guided human motion diffusion model. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Xiaohan Zhang, Sebastian Starke, Vladimir Guzov, Zhensong Zhang, Eduardo Pérez Pellitero, and Gerard Pons-Moll. SCENIC: Scene-aware semantic navigation with instruction-guided control. In *arXiv:2412.15664*, 2024.

Yan Zhang, Yao Feng, Alpár Cseke, Nitin Saini, Nathan Bajandas, Nicolas Heron, and Michael J. Black. PRIMAL: Physically reactive and interactive motor model for avatar learning. In *ICCV*, 2025. arXiv:2503.17544.

Kaifeng Zhao, Gen Li, and Siyu Tang. DartControl: A diffusion-based autoregressive motion model for real-time text-driven motion control. In *International Conference on Learning Representations (ICLR)*, 2025.

Table 2: **Landscape.** The task needs an open horizon *and* metric spatial control *and* typed semantics; each existing line supplies at most two.

Approach (representative work)	Horizon	Metric spatial	Semantics
Sparse spatial control GMD (Karunratanakul et al., 2023), OmniControl (Xie et al., 2024), CondMDI (Cohan et al., 2024), SFControl (Hwang et al., 2025), MaskControl (Pinyoanuntapong et al., 2025)	1 window	**sub-cm–cm**	clip text
Streaming autoregression DART (Zhao et al., 2025), MotionStreamer (Xiao et al., 2025), COMET (Lee et al., 2026)	**open**	seed / per-joint / none	text or none
Scene-bound synthesis Story-to-Motion (Qing et al., 2023), LINGO (Jiang et al., 2024), SCENIC (Zhang et al., 2024)	**open**	waypoint	instruction
Actpin this paper	**open**	**world anchors, sub-cm**	**action per event, as an objective**

Appendix

This appendix gives the reproducibility detail for the studies in the main text: the diagnosis protocol, the embedding vocabulary and the transition (bond) model, the APB's construction, metrics, and per-method inputs, the substrate and deployed settings, the seam and plausibility audits, the full coupling and two-pin tables, the rendered comparisons, and the verbatim benchmark listing.

A Problem definition: details

A TRANSITION τ_k routes the body from site s_k^- to the next site s_k^+, arriving at end frame t_k^e (the per-transition *arrival* term of Eq. 1). The responsible joint is fixed by the action class (the right foot for a kick, the right wrist for a throw; the full class-to-joint map is in Appendix F). $\mathcal{M}$ denotes the physically valid motions: plausible posture, rigid bones, non-sliding foot contact: the axes the validity metrics of Sec. 4.2 score. The horizon T is set by the itinerary, not fixed in advance, which is what rules out any single fixed-window formulation from the start.

B Diagnosis: full protocol and table

Table 3: **Where each demand lives.** Every family already *optimizes* for space at test time or trains for it; none holds the verb in any objective. The two demands sit on different variables at different stages, so when they conflict, space wins by construction; ours is the column where both sit on *one* variable, in one program, scored on the final output.

	semantics enters	space enters	shared variable?	final output re-scored?
guidance (GMD, OmniControl)	text condition	per-step gradient on the noisy sample	no	no
token opt. (MaskControl)	text condition	logits optimization (TTA, bypassed grads)	no	spatial only
imputation (CondMDI)	text condition	observed-keyframe inpainting	no	no
densification (SFControl)	text condition	sparse→dense keyjoints, then completion	no	no
constraint-trained (Kimodo)	text condition	training-time conditioning	no	no
Actpin (ours)	**objective** on rendered frames	**ε-constraint**, same program	**yes: the seeds**	**both pins, every step**

The GT-mined protocol follows OmniControl's home evaluation: anchors are mined from ground-truth clips that perform the commanded verb, the model is conditioned on the caption plus that mined anchor (the "hint"), and success is a met point with a retrievable caption. Table 4 is the full audit summarized in Sec. 2.2. Its last row runs *MaskControl* on the same mined events and separates the two artifacts cleanly: the *leak* transfers in full (token optimization reads .962 on GT-mined anchors against .508 on authored ones, Table 1) while the *snap* does not (accel@t_c $3.5\times$, bone $1.28\times$, both near the text-only range): the leak is a property of the protocol, the snap of gradient guidance.

One artifact deserves the number behind it: much of guidance's remaining world-frame success is a *root shift*, not articulation: the whole body is translated until the joint coincides with the point, the pose barely changing relative to the body (40 cm of body-frame anchor error under 0.5 cm world error). Both breakage axes grow dose-wise as the mined anchor is shifted off its source clip ($148\rightarrow$

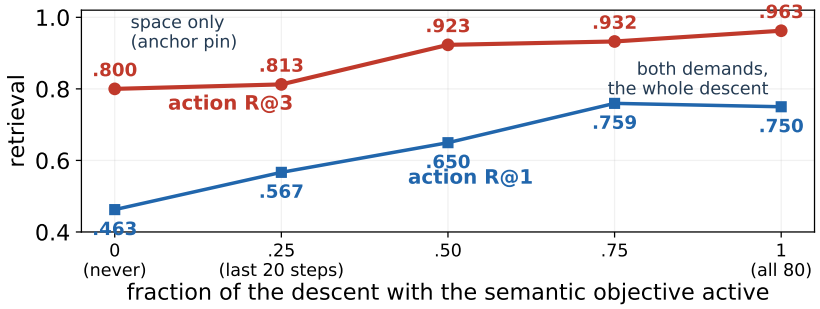

Figure 6: **The timing dose: the two demands must share the descent.** The bare two-pin call-base program with the semantic objective switched on only for the late part of the 80-step descent, everything else fixed (x: fraction of the descent with the objective active; 0 is the anchor-only arm, 1 the bare two-pin arm; tiered APB; anchors hold at 1.2–1.5 cm on every arm). A late verb buys almost nothing (.800 → .813 for the last quarter); the read climbs only as the two gradients coexist (.923 at half), and the strict R@1 keeps rising through three-quarters (.759). Optimizing space first and semantics after is not the same program as optimizing both together: Table 3's *when* column is load-bearing.

Table 4: **Diagnosis on the home protocol: GT-mined anchors** (OmniControl regime; identical commanded verbs; † subset-matched, 8 clips). Accel@t_c = responsible-joint acceleration at the commanded frame over the clip median (med / p90); Bone = worst per-bone stretch inside the reach span over the clip's median bone length (med; a rigid skeleton is 1.0). "Hinted" = conditioned on the mined anchor. The mined anchor *helps* retrieval (.62 → .92) while the body breaks to meet it, and shifting the mined anchor off its clip makes the breakage grow dose-wise while retrieval stays high.

Arm (OmniControl)	n	Ctrl (cm)	R@1↑	R@3↑	Accel@t_c↓	Bone↓
text-only (no hint)†	13	–	.231	.615	3.8 / 9.7	1.03×
hinted (deployed scale)	78	0.5 med	.692	.923	148 / 2002	2.21×
hinted, anchor shifted 1.0 m	78	follows (r=.97)	.744	.897	178 / 1288	2.71×
hinted, anchor shifted 2.0 m	78	follows	.692	.923	327 / 1942	3.39×
The same mined events, token optimization (MaskControl): the leak transfers, the snap does not						
hinted (token optimization)	55	0.4 med	.906	.962	3.5 / 8.0	1.28×

327× acceleration; 2.21 → 3.39× worst bone) while retrieval holds at .90+: the metrics that certify success cannot see how it is achieved. The shifted anchor is *not* met from a standstill: guidance moves the coordinates regardless of the caption, so the root travels with the dose (median net root displacement 0.24 → 0.72 → 1.33 m across the 0/1/2 m shifts; sequences generate root-normalized at the origin, so the start cannot simply be re-seated at the anchor), and the remainder the travel does not cover is what the snap and the stretch close.

C THE EMBEDDING VOCABULARY AND ITS KINEMATIC COUPLING

The conditioning bank is built offline from a frozen 73 M segment-embedding model over 5 body tracks (per-segment embeddings on the 256-d TMR-aligned sphere; a span's embedding is the presence- and length-weighted mean of its member segments, renormalized). Generation consults only the fixed bank: the primitive denoiser and head W are finetuned once to read it in place of text (K=10 sampling steps, classifier-free guidance 2.0). Two measured properties justify it as the c_k of Eq. 2. *First, the embedding is a spatial-conditioning signal, not a label*: ridge regressions from a span's embedding (observables recovered from the raw HumanML3D features; duration-only and embedding-shuffle controls alongside) predict realized speed at R^2=0.94 against ≈0 for both controls, and still 0.85 when the frozen decoder *generates* the schedule (n=2592 locomotion spans; Table 5); swapping DART's CLIP text for the bank on matched rollouts preserves or strengthens every action channel but squat (Table 6). *Second, the transition embedding is predictable from context*: a held-out transition is predicted at cosine 0.67 (context-free vectors ≤ 0.46; a canned "walk" −0.03), and interpretably so, slower into a careful squat and brisk into a dynamic kick (Table 8).

Table 5: **Embedding-space coupling (R^2).** Embedding predicts spatial observables; duration and shuffle controls do not; the coupling reproduces when the decoder *generates* the schedule.

Target	Cont.	Dur.	Shuf.	**Gen.**
Speed	**0.94**	0.00	−0.01	**0.85**
Displacement	0.62	0.21	−0.00	0.56
Cadence	0.53	0.01	−0.01	0.45
Jump rise	0.71	−0.01	−0.06	0.69

Table 6: **Embedding vs. CLIP text on the identical rollout** (per-verb limb channel; means). Embedding ≥ text on every action but squat; no-verb→walk collapses actions.

Cond.	Run↑	Kick↑	Wave↑	Punch↑	Squat↑
text	1.15	−.11	.56	.65	**.52**
embedding	1.75	**.09**	**.63**	**.73**	.49
→walk	**1.89**	−.16	.45	.71	.51

Table 7: **Heading-seam fix** (path-leg curvature, 1 =straight). The heading-realignment step straightens the large-turn legs (those inheriting a heading more than 40° off the goal) at negligible anchor cost.

	turn ≥40°	all legs
no fix	1.37	1.25
+fix	**1.04**	**1.04**

Table 8: **Transition embedding (bond) is context-determined**: a small model predicts a held-out transition's own 256-d embedding (cosine to ground truth) from the flanking action embeddings and the route geometry, where no context-free vector can.

Predict transition embedding from	**bond**	embeddings	geometry	mean	"walk"
held-out cosine↑	**0.67**	0.47	0.45	0.46	−0.03

D TRANSITION-BOND PREDICTOR

This is the conditioning detail of Sec. 3.4 (Eq. 9): the *bond* predicts a between-reaches locomotion span's 256-d embedding from its context: preceding and following action embeddings and the route geometry (site displacement, span duration, heading change). It is a 3-layer MLP (Linear → SiLU → Linear → SiLU → Linear, hidden 512) that regresses the transition span's 256-d embedding, unit-normalized onto $\mathbb{S}^{255}$, from the two flanking action embeddings, the four route-geometry scalars, and a walk/run/hop selector (a one-hot over the three locomotion modes the model was trained with, equivalent to picking their embedding vector; the benchmark commands all three, with walk and run on most legs and hop on a minority). Trained with a cosine loss (Adam, 300 epochs) on locomotion spans lying between two labeled reaches, clip-split into train and held-out sets. The score is the cosine of predicted-to-true embedding on held-out clips against three context-free baselines: the mean transition embedding, a canned "walk" vector, and the two single-context ablations (embeddings-only, geometry-only). At inference the predicted bond embedding conditions the goal-reaching controller for the transition segment in place of a fixed locomotion vector.

E HEADING REALIGNMENT

Chaining leaves one artifact: a leg that inherits a heading far off the next goal realizes the required turn as a wide arc that overshoots. While the heading is over 40° off the goal bearing we feed π_θ a temporary near goal on the true bearing, dropping it into its trained near-goal regime where it pivots almost in place, then restore s^+; every frame stays policy-generated, and only sharp-turn legs are touched (curvature 1.37 → 1.04 on legs over 40°, Table 7).

F APB CONSTRUCTION, METRIC DEFINITIONS, AND PER-METHOD INPUTS

Construction. From the origin, sites at 2.2–2.6 m force a locomotion segment before each reach; each reach places a class-typical 3D offset along the arrival heading, and the class fixes the responsible joint (HumanML3D index): kick / stomp → r-foot (11), jump / squat → pelvis (0), kneel → r-knee (5), bow → head (15), and the seven hand verbs → r-wrist (21). The core frame is the reach-span midpoint; a leg whose spacing implies >1.0 m/s is commanded as a run, which is why tier 3 (183–190 f) is run-paced by construction. All runs are single-stream; the full listing is Table 17

(Appendix O). **Timing and metrics.** Every method is scored at its own plan's core frame (for the fixed-window baselines the realized span *is* the commanded span; the stream re-times, so it is scored at the core its plan records), and semantics is read over each method's realized span. The stream's re-timing is measured, not hidden: realized cores lag the commanded ones by $+26$ frames median ($+37$ mean, p90 85) because reaches take their natural duration, order and spacing preserved; the itinerary's frame column therefore acts as ordering and pacing, and the baselines, whose fixed window realizes the commanded frame exactly, are read at that frame. Metric definitions are in Sec. 4.2; appendix-level notes: the deployed per-tier anchor means stay sub-centimetre ($0.66/0.74/0.74$ cm over three draws; Table 9); *seam*, the pelvis-speed jump at a join over the clip's own interior jump, is calibrated against the GT action-entry band in Appendix I; and semantics is scored per span rather than by whole-motion FID, which is blind to per-frame breakage (Appendix J).

Why TMR is the reader, and the circularity question. The standard whole-clip evaluator cannot read the span: the Guo text–motion model (Guo et al., 2022) ranks the 13 class captions perfectly on the full mined source clips (R@3 1.00, n=25) but collapses on 44-frame crops of the same clips (.68) and is at chance on every method's generated spans (.22–.35; chance .23); TMR is the one retrieval model that still separates methods at the task's span length. The optimize-and-judge concern is bounded by construction: the optimization target is a real-clip *motion-encoder* prototype while the score ranks the class captions' *text-encoder* embeddings; a non-TMR target still reads .753 (Table 14); and the anchor-only arm, with no semantic term at all, reads .800. Bootstrap 95% CIs (10^4 resamples over the 240 reaches): deployed R@3 .913 [.875, .946], R@1 .696 [.637, .754]; the two-pin gap over anchor-only is [.06, .16] R@3, over the strongest baseline [.34, .47].

What each method receives. Every itinerary also carries natural-language captions (a caption per event and a whole-itinerary caption), and no baseline is fed the table's raw form; each receives the itinerary translated into its own native interface, the most favourable encoding we could construct:

- *OmniControl / MaskControl* (guidance / token control): the window's caption as text, plus sparse joint targets, the responsible joint's world anchor at its core frame and the site as a pelvis keyframe at each transition's end frame.
- *CondMDI / SFControl* (imputation / keyjoint densification): the same caption and the same sparse keyframes, injected as observed frames (CondMDI) or sparse keyjoint constraints (SFControl).
- *Kimodo*: the anchors as end-effector contact constraints with their core-frame timing.
- *Ours*: the typed events exactly as authored, the captions unused at generation (embedding comes from the class bank and the bond).

Two baselines read far below their published numbers, and we verified both pipelines against their own regimes. *SFControl*: feeding its stage 2 ground-truth *dense* keyjoints reproduces the paper's sub-centimeter figure (0.8 cm), while its own sparse protocol on the HumanML3D test split yields 24 cm, our GT-mined probe 48 cm, and authored anchors 99 cm: the degradation is monotone in how far the control signal sits from the densifier's training manifold. *Kimodo*: its shipped end-effector tutorial runs at 2–4 cm under our installation, but its constraint format in practice is a motion fragment (frame-indexed local rotations plus root positions saved from an existing motion), and a bare free-space point through its end-effector channel sends the root drifting by meters (the same symptom is reported on the released model in the authors' own issue tracker, open at submission time; coordinate conventions were verified end to end, and the accepted root waypoints land at 4 cm). The APB therefore gives it its strongest applicable encoding, root waypoints plus per-event text, which lands the body without pinning the joint (70 cm). Neither system claims authored free-space anchors, which is the point of the audit: nothing in the field does.

Beyond every baseline's window (tier 4–5 probe). The APB stops at three actions only so that every itinerary fits one native window; the stream has no such limit. Twenty extension itineraries authored by the same rules (four or five actions each, 90 reaches, 243–315 f) run the program unchanged (the ε=1, site-free variant): anchors 1.48 cm, hand-over in the band (3.1×), and R@3 .950 at tier 4, .900 at tier 5, so the verb survives at horizons no fixed-window baseline can enter.

Table 9: **The difficulty axis: per-tier anchor and read** (tier = actions per itinerary; every tier fits one window, so difficulty is *composition alone*). Anchor = per-tier *mean* (cm), so each row's reach-weighted mean over tiers equals its Table 1 anchor (ours = three-draw means, like Table 1). Composing more actions leaves ours high on tiers 1–2 and lower on tier 3 (.858–.875 over draws); the strongest placer's verb collapses (.73 → .43), and guidance loses even its *placement* (4.8 → 37 cm), inside one window.

	Anchor (cm)↓			R@3↑		
Method	t1	t2	t3	t1	t2	t3
OmniControl	4.8	16.4	37.0	.675	.425	.333
MaskControl	0.7	1.4	1.8	.725	.512	.433
CondMDI	65.6	68.8	77.9	.575	.325	.233
SFControl	177.5	106.5	68.3	.525	.388	.333
Kimodo	65.0	70.4	71.6	.250	.237	.242
Ours (deployed)	0.66	0.74	0.74	**.942**	**.942**	**.864**

G SUBSTRATE DETAILS: ROLLOUT AND TRAVEL POLICY

The stream of Eq. 2 re-canonicalizes its history each step, $h_{k+1} = \mathrm{canon}(x_{k+1})$, and its condition c_k is the current event's embedding: the class embedding on a reach, the bond embedding $\hat{g}$ on a transition. On a travel primitive the seed is written by a goal-reaching controller trained by reinforcement to drive the same denoiser,

$$a_k = \pi_\theta(o_k), \qquad o_k = \big(\widehat{s^+ - p_k},\ \|s^+ - p_k\|,\ W(\hat{g}),\ h_k,\ \eta_k\big), \tag{10}$$

where $\widehat{v}$ is the unit vector, p_k the pelvis ground position, W the *same* head the denoiser reads (the controller sees the bond conditioning the denoiser sees), and η_k a step counter. Because networks and history are shared across every primitive, run-up momentum carries across each transition-to-reach boundary and timing is emergent, each span running until the controller arrives; a heading-realignment step handles sharp inherited turns (Appendix E). The controller is DART's goal-reaching PPO recipe retrained once on the embedding-conditioned denoiser. A residual travel artifact is a near-goal *orbit* on a minority of legs (12–16% exceed 1.5× path curvature): the policy overshoots a close goal and arcs back. An anti-orbit variant of the realignment (refreshing the pivot budget on aligned exits and forcing a pivot on a near-goal overshoot) trims the worst loops (max curvature $5.7 \to 3.7\times$) but costs .017 R@3 on the benchmark, so it ships as an optional flag rather than the default.

H ACTPIN SEEDING: DEPLOYED SETTINGS AND COST

Spatial pin. Seeding optimizes the reach burst's k primitive seeds jointly, once per reach: 80 Adam steps at learning rate 0.05 decayed linearly to 0, each step re-rendering the burst through the frozen $G, \mathcal{D}$ with per-seed L_2-normalized gradients; this short, decaying budget realizes the trust region of Eq. 6. Eq. 7 deploys floor 0.3, velocity-continuity λ_v=20, and pelvis-stay 2.0 (gated off for squat, kneel, pick); the anchor-pin-only ablation replaces the priced constraint with a Huber anchor term. The embedding conditions *every* primitive of the reach (core-only conditioning starves the wind-up and recovery, measured on the ε=1 variant: R@3 .942 → .925, R@1 .758 → .700; a ±8-frame critic window is far worse, .833). Denoising uses K=10 steps at guidance 8 on reaches and 5 on travel.

Arrival extension (deployed). The program rewinds the last travel primitive before each reach and co-optimizes its seed with the burst (*trans-opt*; boundary pelvis-speed smoothness 100), with the *site held as a second ε-constraint* on the rewound primitive's end pelvis ($\varepsilon_{\text{site}}$=0.5 cm, one closed-form multiplier per constraint in the same dynamic barrier) and the anchor tolerance at ε=0.5 cm: both spatial demands are constraints, semantics stays the only objective, and this drives arrival to 0.5 cm and anchors to 0.81 cm at a −.029 R@3 cost against the ε=1 site-free variant (Table 14). The constraint form dominates the common practice of pricing the site as a fixed-weight penalty: the penalty frontier w=20/40/60/80 reads arrival 3.0/1.3/0.9/0.6 cm at R@3 .938/.908/.904/.858, reaching sub-centimetre arrival only where the read pays more than the constraint form pays at 0.5 cm (.913). A hybrid that keeps the anchor barrier but handles the site by classical ALM reaches the same arrival (0.5) while losing the anchor (0.81 → 1.35 cm, p90 2.0): the accumulated site price overwhelms the anchor's memoryless one, so *per-constraint* barrier multipliers are what hold both.

Table 10: **Boundary metrics, all four views** (medians over the 240 joins). PJ/AUJ: FlowMDM (Barquero et al., 2024), 1 s window on the join (m/frame^3). Ratio = our pelvis-speed diagnostic (boundary speed jump over the clip's interior median $|\Delta v|$); Abs = the un-normalized jump (cm/frame). GT row: real action entries (n=63–73). PJ/AUJ certify no positional tear for any stream arm; only the ratio exposes the bare program's rhythm break (and the anchor-only arm's missing entry), which is why we deploy the arrival extension.

	PJ@join	AUJ@join	Speed ratio (→GT band)	Abs (cm/f)
GT action entries	.22	1.31	2.6 (1.2–3.9)	0.34
OmniControl	1.64	4.59	2.3	0.61
MaskControl	.12	.64	3.5	0.59
CondMDI	.21	1.97	0.3	0.74
SFControl	.33	3.71	1.3	–
Kimodo	11.6	36.4	355	–
unsteered stream	.12	.64	4.6	–
anchor pin only	.16	.73	0.1	–
two-pin (− arrival ext.)	.15	.73	8.0	1.82
full program (deployed)	.14	.69	4.2	0.97

A looser site ball ($\varepsilon_{\text{site}}$=1 cm) buys read (.925/.741) at arrival 1.0 cm, and loosening the *anchor* ball instead (ε=1 cm, site kept at 0.5) buys it at the anchor (.929/.721, anchor 1.19 cm): both are one dial, and we deploy both tight. A sweep ($\lambda_v \in \{10, 20, 40\} \times$ trans-opt) picks λ_v=20; tripling the smoothness weight recovers nothing further, so the deployed residual is the action's own entry dynamics.

Cost. On one RTX A6000, generation runs at 0.46 s per output frame, ∼14× slower than real time: a mean benchmark itinerary (∼320 realized frames, 10.7 s of motion) takes ∼2.5 minutes, dominated by the reach bursts (each of its two reaches re-renders its burst once per descent step, 80 times). Nothing is trained or cached per itinerary. The controller π_θ is DART's goal-reaching PPO recipe retrained once on the embedding-conditioned denoiser (2,000 iterations).

I Boundary quality: standard jerk metrics, and a stricter rhythm diagnostic

The standard metrics. We score every travel→reach boundary with FlowMDM's (Barquero et al., 2024) Peak Jerk and AUJ, computed over a 1 s window on the join (our calibration window), calibrated on the same windows of real action entries (GT PJ .22, IQR .11–.37; AUJ 1.31). Table 10: no stream arm ever tears, the frozen decoder guarantees sub-GT jerk for every seed choice (.12–.16), while guidance spikes (1.64) and Kimodo teleports (11.6).

The rhythm diagnostic. The failure that motivated the arrival extension is invisible to jerk: positions stay smooth while the *gait rhythm* breaks. We therefore also report the boundary jump of pelvis ground speed over the clip's *own* interior median $|\Delta v|$; the normalization is ours, and the denominator matters (run-paced legs are near-constant speed, and a jittery interior shrinks the ratio).

Calibration: the target is the GT band, not zero. Real entries read 2.6× (IQR 1.2–3.9): a wind-up *is* a speed change at exactly that frame. The axis is two-sided: far above the band is a hand-over break (the bare two-pin's 8.0×), far below a *missing* entry (anchor-only 0.1×; CondMDI 0.3× at a .321 read). The deployed program sits just above the band (4.2×; 4.2–4.6 over draws; the light site penalty w=20 sits inside at 3.8× and the ε=1 site-free variant at the band's centre, 2.4×), and on the absolute axis its jump (0.97 cm/frame) sits between MaskControl's (0.59) and the bare program's outright loss (1.82). Driving the ratio further down degrades placement without gaining anything real (Appendix H); what remains is what real performances also show.

J Physical plausibility: where the anchor metric is silent

A low anchor error does not by itself certify a plausible body. Table 11 measures how each anchor is *bought*, one code path for every method, calibrated on GT HumanML3D test clips: bone rigidity (per-bone length CoV over time), worst in-reach stretch, and the GMD and PhysDiff skate metrics (Karunratanakul et al., 2023; Xie et al., 2024; Yuan et al., 2023). Each family games a different axis, which is why no single validity metric suffices: MaskControl's placement comes

Table 11: **Physical plausibility on the APB** (medians over 120 itineraries; one code path for all). A sub-centimetre anchor can be bought with an implausible body, and each family games a different axis: MaskControl stretches, OmniControl snaps, Kimodo slides. GT row: 300 HumanML3D test clips through the same code (calibration).

Method	Anchor↓	Arrival↓	Bone %↓	Worst stretch↓	Skate↓	Skate_{PD} ↓	Span-FID↓
GT (HumanML3D test)	–	–	0.0	1.0×	7.4	0.69	0.98
MaskControl (Pinyoanuntapong et al., 2025)	1.5	1.1	4.9	1.22×	16.8	2.13	20.3
OmniControl (Xie et al., 2024)	24.8	12.8	3.4	1.79×	**12.6**	**1.84**	30.9
CondMDI (Cohan et al., 2024)	72.8	25.7	1.5	1.07×	66.5	3.60	17.4
Kimodo (Rempe et al., 2026)	70.1	6.9	**0.4**	**1.00**×	16.5	47.0	66.3
Ours	**0.81**	**0.5**	2.5	1.18×	13.6	2.48	23.1

with the loosest skeleton of the placers (bone 4.9% vs. our 2.5), OmniControl's snap shows as the largest worst-bone stretch (1.79×), and Kimodo is nearly perfectly rigid yet *teleports* on contact (47 cm/frame toe slide). Ours decodes through a frozen shape-guaranteeing VAE, the most rigid of the anchor-hitting methods at baseline-level skate. For ballistic foot actions the benchmark constrains the span-midpoint core frame, not the impact's own velocity peak; forcing the two to align with a speed floor trades the scored anchor (1.1 → 11 cm), a Pareto move we do not take. *Span-FID* adds the distribution-level axis the per-frame metrics cannot see: the standard text-matched FID needs paired GT motions, which authored itineraries do not have, so we embed every method's realized reach spans with the Guo motion encoder (Guo et al., 2022) and compute FID against 240 length-matched center crops of real test clips (real-vs-real crops read 0.98; the unsteered stream, 13.0, bounds what steering can hope for). Ours sits by the strongest placer (23.1 vs. MaskControl's 20.3, at twice-finer anchors and matching arrival; the penalty-site variants read 20.9–22.6 and the ε=1 site-free variant, 18.9, is the closest of all); CondMDI sits low (17.4) by generating natural motion that misses the task, and the families that break the body pay on this axis too (OmniControl 30.9, Kimodo 66.3).

K THE SEEDING ABLATION, RENDERED

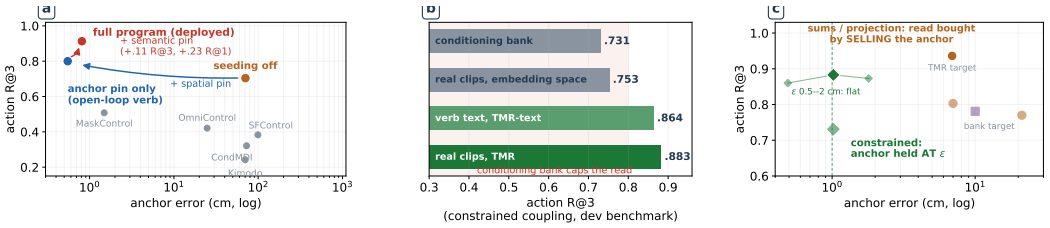

Figure 7: **actpin seeding, taken apart.** *(a)* The loop delta on the APB: seeding off misses outright, the spatial pin alone places but leaves the verb open-loop, the deployed program holds both (0.81 cm, .913; grey: the baselines of Table 1). *(b)* The semantic target must live in the reader's space (dev benchmark). *(c)* The coupling must be constrained, not summed: sums buy read by *selling* the anchor, the constrained program holds ∼1 cm at the strongest read. Full tables: Tables 16 and 14.

Table 12: **The price-law menu against the setting's requirements** (R1 moving landscape, R2 one-sided violation, R3 short budget; Sec. 3.3). Memory is the discriminating property; the study of Sec. 4.4 tests this prediction.

Price law	update for λ	memory	predicted behavior under R1–R3
weighted sum	fixed w	(no constraint)	either pin can buy the other
projection	$\nabla f \perp \nabla g$	(no constraint)	per-step orthogonality drifts off the level set
first-order primal–dual (Arrow et al., 1958)	$\lambda{+}\mu g$ / step	integral	stale λ keeps fighting the verb
PID-Lagrangian (Stooke et al., 2020)	$+K_p\Delta g{+}K_d\Delta^2 g$	integral	same, damped
classical ALM (Hestenes, 1969)	$\lambda{+}\mu g$ / outer loop	integral	same, slower to stale
quadratic penalty (Courant, 1943)	$\mu_t\ g$, μ_t on a schedule	memoryless	releases; gain rides the schedule
dynamic barrier (Gong et al., 2021)	closed-form $\lambda^\star$ (Eq. 8)	memoryless	releases; gain rides geometry

Table 13: **The coupling design study at the deployed layer**: the five price laws of Table 12, plus the signed primal–dual variant, each running the deployed two-constraint program (anchor and arrival site, one multiplier per constraint) under one matched draw (seed 42, like Fig. 5c), identical soft terms and budget; 240 reaches per law; anchor = mean (cm). The anchor separates nothing; the *arrival* ball does: slow-integrating prices never bind it within the step budget. **Deployed = the dynamic barrier**. *one span non-finite (the signed variant may *unwind* inside the ball).

Price law	Anchor↓	Arrival↓	R@3↑	R@1↑
first-order primal–dual	**0.52**	1.4	.888	.629
+ signed violation*	0.62	2.2	.887	.634
classical ALM (1969)	0.65	**0.5**	.892	.646
PID-Lagrangian (2020)	0.58	1.7	.888	.621
quadratic penalty (1943)	0.61	**0.5**	**.929**	.692
dynamic barrier (2021)	0.81	**0.5**	.913	**.696**

L THE TWO PINS, TAKEN APART (FULL TABLE)

Table 14 is the full two-pin ablation summarized in Sec. 4.5: the target and coupling axes, the ε-frontier, and the robustness rows on the tiered APB. The deployed program is draw- and split-stable: the deployed point reads .913/.896/.900 over three stream draws (Table 1 reports their mean) and .921 on a fully re-authored seed-disjoint split, with arrival at 0.5 cm on all four (the constraint binds identically on every draw; anchors 0.68–0.81 cm, boundary jump 4.2–4.6×), and the ε=1 site-free variant adds four draws within .022 R@3; the loop, not a lucky draw or a memorized layout, carries the result. Two further one-term dials were swept and left at their defaults: raising content guidance to 12 buys only draw noise (.958 on its draw, .921 on a redraw; guidance 5 loses), and a small bone-rigidity term (w=3) halves the transient spike tail for −.017 R@3; the operating point is a choice, not a cliff.

Table 14: **The two pins, taken apart.** Top: target and coupling axes on the dev benchmark (core-primitive conditioning, predating the coverage study of Appendix H). Bottom: the deployed whole-reach program on the tiered APB (n=240; speed ratio = median boundary jump). ‡ seed 42.

Semantic target	Coupling	Anchor↓	Speed ratio (→GT)	R@3↑	R@1↑
(none: anchor pin only)	–	0.81	0.6	.665	.330
class bank (embedding)	weighted sum w=.2	7.0	4.0	.803	.430
class bank (embedding)	constrained	1.02	5.2	.731	.400
real clips (embedding)	constrained	1.00	4.7	.753	.484
verb text (TMR-text enc.)	constrained	0.98	4.5	.864	**.660**
real clips (TMR)‡	**constrained = two-pin**	**1.02**	4.9	**.883**	.547
re-verified on the tiered APB (whole-reach conditioning)					
two-pin, closed loop (− arrival ext.)‡		1.34	8.0	**.963**	**.750**
deployed (anchor + site both ε=0.5 cm; arrival 0.5)‡		0.81	4.2	.913	.696
stream redraw, seed 43 (arrival 0.5)		0.68	4.2	.896	.746
held-out re-authored split (arrival 0.5)		0.69	4.4	.921	.700
stream redraw, seed 44 (arrival 0.5)		0.69	4.6	.900	.688
anchor ball ε=1 cm, site $\varepsilon_{\text{site}}$=0.5 (arrival 0.5)‡		1.19	4.4	.929	.721
site ball $\varepsilon_{\text{site}}$=1 cm (arrival 1.0; 1/240 invalid)‡		0.80	4.3	.925	.741
site by classical ALM (anchor pays; arrival 0.5)‡		1.35	4.3	.921	.688
site penalty w=60 (prior deployed; arrival 0.9)‡		0.82	4.4	.904	.696
its redraws/split (arrival 0.9–1.0)		0.71–0.89	4.1–4.3	.929–.933	.700–.738
site penalty w=20 (arrival 3.0; redraw/split .949/.933)‡		0.78	3.8	.938	.746
ε=1, site-free (arrival 7.7)‡		1.10	**2.4**	.942	.758
four draws (seeds 42–45)		1.10–1.19	2.3–3.2	.924–.946	.718–.758
ε=0.1/0.5/2 cm, site-free‡		0.29/0.65/2.05	2.7–3.3	.908–.933	.717–.771

The reach-up target: a prototype–critic mismatch, found and fixed. The class prototype for *reach up high* was nearly orthogonal to the caption embedding the benchmark scores against ($\cos 0.078$), pushing reaches toward "throw"; the deployed target set replaces this one prototype with the caption's own TMR-text embedding (class R@3 .727 → 1.000 at intact anchors). All other classes keep their real-clip prototypes, and the coupling study applies the fixed set to every law alike.

Residual failure modes, and two targeted extensions. The barrier still misses in two ways: 5/109 traced reaches stall short of the ε-ball, and released iterates transiently re-violate after capture

(24/80 steps, median). An anticipatory margin (δ=4 mm) or a conditioning cap on λ^* fixes each (capture $\rightarrow$98%, re-violation $\rightarrow$10 steps), but neither improves the *read*: extra conservatism buys constraint robustness at the price of tail retrieval, so we deploy the plain barrier and keep both terms as dials.

M PER-FAMILY ACCELERATION AUDIT: AUTHORED VS. GT-MINED ANCHORS

Table 15 dissects *how* each family meets its anchors, per reach event, on both anchor regimes: the top block runs every baseline on the APB's authored anchors (the deployment case of Sec. 4.3) and reads the responsible joint's acceleration at the commanded frame; the bottom block puts ours back on the GT-mined events of the diagnosis (Sec. 2.2), closing the loop with their own protocol.

Table 15: **Per-event dissection.** Top: the same controllers on the *authored* anchors of the APB (covered reaches); Accel@t_c = responsible-joint acceleration at t_c over the clip median (med / p90 / %>2×). Bottom: ours, back on the *GT-mined* events of Table 4 (their protocol). $*$ impact verbs naturally exceed 2×; *magnitude* separates snap from motion.

Arm	n	Ctrl (cm)	R@1↑	R@3↑	Accel@t_c (med/p90)↓	%> 2×
Authored anchors (APB; covered reaches of 240*)*						
OmniControl	206	13.4	.226	.486	45.2 / 114	100
MaskControl	206	1.1	.264	.557	2.7 / 7	70
CondMDI	206	66.7	.198	.396	**0.9** / 2	3
SFControl	232	111.0	.192	.385	1.8 / 4	44
Kimodo	208	67.7	.072	.249	5.5 / 14	88
Ours on the GT-mined events (OmniControl protocol, Sec. 2.2; core-primitive conditioning, predating Appendix H's coverage study)						
ours, two-pin (80-step budget)	24	1.1 med	.625	.917	8.3 / 18	100*
ours, two-pin (converged 160)	24	1.19 mean	**.958**	**1.00**	10.7 / 26	96*

N COMPONENT ABLATION, RENDERED

Table 16: **Stream-component ablation** (tiered APB, 240 reaches, equal policy and seed 42; Table 1 reports the deployed three-draw mean). Seeding carries the anchor axis: with the steering channel off the responsible joint lands 71 cm off; the anchor pin alone places but leaves the verb open-loop. (An earlier draft carried a learned transition-embedding head; removing it is measurably free, so it is no longer part of the method. The heading realignment's contribution is quantified in Appendix E.)

Condition	Anchor (cm)↓	R@3↑	R@1↑
full − seeding (unsteered)	70.6	.704	.450
anchor pin only (open-loop verb)	**0.55**	.800	.463
two-pin, closed loop (− arrival ext.)	1.34	**.963**	**.750**
full program (deployed)	0.81	.913	.696

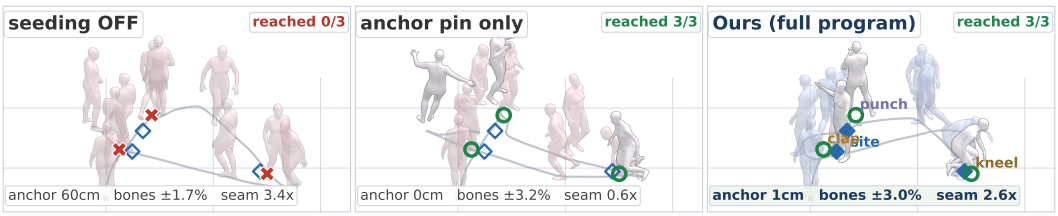

Figure 8: **The closing-the-loop delta, rendered** (APB spec 95, shared camera; green ring = reached, red cross = missed; blue diamond = travel site, filled = pelvis route passes within 1 cm). Left: with seeding off, the same stream renders the same events but no anchor is met. Middle: the anchor pin alone places every joint, but the verb has no defender, and the spans read as generic motion. Right: the deployed full program meets every anchor in a legible posture. Benchmark means in Tables 1 and 16.

O APB LISTING

Table 17 lists the Action Pinning Benchmark (Sec. 4.2) verbatim: all 120 tiered itineraries (40 per tier of $1/2/3$ actions, 240 reaches).

How a listed row enters each method (worked example, spec 15). The row "T walk, R reachup @75 $(-0.78, 2.05, -2.44)$" is delivered as follows. *Ours* receives the typed events verbatim: {TRANSITION: walk, site $(-0.75, -2.34)$, frames 0–56}, then {REACH: reach-up, responsible joint r-wrist (21), anchor $(-0.78, 2.05, -2.44)$, core frame 75}; the captions are unused at generation. *Every baseline* receives the same events translated to its native interface (per-family mapping in Appendix F): the text prompt is the itinerary caption, here "*a person walk forward then reach up high*" (per-event captions "*walk forward*", "*reach up high*"), plus the sparse spatial signals, the r-wrist anchor as a joint target at frame 75 and the site $(-0.75, -2.34)$ as a pelvis keyframe at the transition's end frame 56.

Table 17: **The Action Pinning Benchmark, verbatim** (all 120 itineraries; tier = actions per itinerary; frame indices at 30 fps; T=transition with its locomotion type, R=reach with responsible-joint core frame and 3D world anchor in metres; the italic line is the itinerary's text caption, the prompt every baseline receives).

Spec	Tier	Len.	Ordered events (transition; reach @ core, anchor)
0	1	92 f	T walk, R kick @71 (-1.36,0.55,2.3) *"a person walk forward then kick"*
1	1	111 f	T hop, R jump @89 (1.47,1.05,2.28) *"a person hop forward then jump forward"*
2	1	62 f	T run, R push @46 (2.89,1.2,-0.3) *"a person run forward then push forward"*
3	1	88 f	T walk, R pick @71 (0.98,0.15,-2.57) *"a person walk forward then pick something up"*
4	1	92 f	T walk, R bow @73 (-1.85,0.95,-2.03) *"a person walk forward then bow"*
5	1	96 f	T walk, R stomp @76 (-2.54,0.1,0.7) *"a person walk forward then stomp"*
6	1	117 f	T hop, R kneel @95 (-0.51,0.1,2.58) *"a person hop forward then kneel down"*
7	1	61 f	T run, R throw @45 (2.26,1.6,1.77) *"a person run forward then throw a ball"*
8	1	92 f	T walk, R clap @75 (2.54,1.35,-1.2) *"a person walk forward then clap hands"*
9	1	86 f	T walk, R punch @68 (0.08,1.3,-2.75) *"a person walk forward then punch"*
10	1	90 f	T walk, R squat @70 (-1.98,0.55,-1.07) *"a person walk forward then squat down"*
11	1	110 f	T hop, R push @89 (-2.37,1.2,1.65) *"a person hop forward then push forward"*
12	1	74 f	T run, R pick @52 (0.38,0.15,2.71) *"a person run forward then pick something up"*
13	1	86 f	T walk, R bow @70 (2.58,0.95,0.88) *"a person walk forward then bow"*
14	1	90 f	T walk, R stomp @72 (1.87,0.1,-1.83) *"a person walk forward then stomp"*
15	1	94 f	T walk, R reachup @75 (-0.78,2.05,-2.44) *"a person walk forward then reach up high"*
16	1	116 f	T hop, R throw @95 (-2.82,1.6,-0.45) *"a person hop forward then throw a ball"*
17	1	73 f	T run, R clap @51 (-1.64,1.35,2.27) *"a person run forward then clap hands"*
18	1	90 f	T walk, R punch @74 (1.44,1.3,2.79) *"a person walk forward then punch"*
19	1	85 f	T walk, R squat @68 (2.23,0.55,-0.02) *"a person walk forward then squat down"*

Table 17 continued

Spec	Tier	Len.	Ordered events (transition; reach @ core, anchor)
20	1	89 f	T walk, R kick @70 (1.2,0.55,-2.44) *"a person walk forward then kick"*
21	1	109 f	T hop, R pick @89 (-1.63,0.15,-2.17) *"a person hop forward then pick something up"*
22	1	72 f	T run, R bow @51 (-2.67,0.95,0.48) *"a person run forward then bow"*
23	1	85 f	T walk, R stomp @70 (-0.74,0.1,2.49) *"a person walk forward then stomp"*
24	1	89 f	T walk, R reachup @72 (1.85,2.05,1.75) *"a person walk forward then reach up high"*
25	1	93 f	T walk, R wave @75 (2.35,1.55,-1.11) *"a person walk forward then wave"*
26	1	114 f	T hop, R clap @94 (0.34,1.35,-2.76) *"a person hop forward then clap hands"*
27	1	71 f	T run, R punch @50 (-2.6,1.3,-1.73) *"a person run forward then punch"*
28	1	95 f	T walk, R squat @73 (-1.93,0.55,1.09) *"a person walk forward then squat down"*
29	1	84 f	T walk, R kick @68 (0.13,0.55,2.7) *"a person walk forward then kick"*
30	1	88 f	T walk, R jump @70 (2.51,1.05,1.12) *"a person walk forward then jump forward"*
31	1	107 f	T hop, R push @88 (2.28,1.2,-1.86) *"a person hop forward then push forward"*
32	1	70 f	T run, R stomp @50 (-0.55,0.1,-2.52) *"a person run forward then stomp"*
33	1	99 f	T walk, R reachup @77 (-2.45,2.05,-0.63) *"a person walk forward then reach up high"*
34	1	87 f	T walk, R wave @71 (-1.5,1.55,2.1) *"a person walk forward then wave"*
35	1	91 f	T walk, R kneel @74 (1,0.1,2.47) *"a person walk forward then kneel down"*
36	1	112 f	T hop, R throw @94 (2.9,1.6,0.24) *"a person hop forward then throw a ball"*
37	1	69 f	T run, R squat @49 (1.36,0.55,-2.21) *"a person run forward then squat down"*
38	1	93 f	T walk, R kick @72 (-1.41,0.55,-2.29) *"a person walk forward then kick"*
39	1	82 f	T walk, R jump @67 (-2.72,1.05,0.23) *"a person walk forward then jump forward"*
40	2	185 f	T walk, R wave @76 (-0.34,1.55,2.46), T walk, R kneel @166 (1.71,0.1,2.96) *"a person walk forward then wave then walk forward then kneel down"*
41	2	175 f	T walk, R punch @71 (2.55,1.3,-1.07), T walk, R reachup @160 (3.45,2.05,1.11) *"a person walk forward then punch then walk forward then reach up high"*
42	2	157 f	T run, R squat @50 (0.15,0.55,-2.23), T walk, R reachup @139 (-0.52,2.05,-4.52) *"a person run forward then squat down then walk forward then reach up high"*
43	2	162 f	T walk, R pick @70 (-2.26,0.15,1.44), T run, R bow @140 (-0.97,0.95,-1.15) *"a person walk forward then pick something up then run forward then bow"*
44	2	174 f	T walk, R reachup @70 (0.24,2.05,2.39), T walk, R bow @158 (-1.56,0.95,4.33) *"a person walk forward then reach up high then walk forward then bow"*
45	2	175 f	T walk, R wave @69 (2.35,1.55,0.64), T walk, R push @157 (4.71,1.2,-0.61) *"a person walk forward then wave then walk forward then push forward"*
46	2	132 f	T run, R clap @46 (1.92,1.35,-1.74), T run, R push @112 (-1.1,1.2,-1.04) *"a person run forward then clap hands then run forward then push forward"*
47	2	182 f	T walk, R squat @77 (-2.34,0.55,-0.47), T walk, R kick @165 (-4.69,0.55,-1.71) *"a person walk forward then squat down then walk forward then kick"*
48	2	159 f	T walk, R push @71 (-1.74,1.2,2.21), T run, R kick @140 (0.55,0.55,3.79)

Table 17 continued

Spec	Tier	Len.	Ordered events (transition; reach @ core, anchor)
			"a person walk forward then push forward then run forward then kick"
49	2	186 f	T walk, R pick @71 (1.11,0.15,2.38), T walk, R punch @165 (-0.22,1.3,-0.6) *"a person walk forward then pick something up then walk forward then punch"*
50	2	151 f	T run, R reachup @49 (2.35,2.05,0.07), T walk, R punch @136 (2.48,1.3,2.89) *"a person run forward then reach up high then walk forward then punch"*
51	2	170 f	T run, R wave @49 (0.92,1.55,-2.2), T hop, R throw @152 (1.73,1.6,-4.5) *"a person run forward then wave then hop forward then throw a ball"*
52	2	173 f	T hop, R clap @86 (-1.44,1.35,-2.09), T run, R throw @153 (-2.71,1.6,0.24) *"a person hop forward then clap hands then run forward then throw a ball"*
53	2	184 f	T walk, R punch @70 (-2.84,1.3,0.39), T walk, R wave @162 (-0.48,1.55,-1.37) *"a person walk forward then punch then walk forward then wave"*
54	2	171 f	T walk, R jump @69 (-0.91,1.05,2.63), T walk, R wave @155 (-2.98,1.55,3.2) *"a person walk forward then jump forward then walk forward then wave"*
55	2	151 f	T walk, R push @69 (2.06,1.2,2.12), T run, R wave @133 (4.03,1.55,1.48) *"a person walk forward then push forward then run forward then wave"*
56	2	151 f	T run, R stomp @45 (2.46,0.1,-0.77), T walk, R stomp @131 (-0.02,0.1,-1.56) *"a person run forward then stomp then walk forward then stomp"*
57	2	156 f	T walk, R wave @72 (-2.02,1.55,-1.17), T run, R pick @139 (-3.23,0.15,-3.66) *"a person walk forward then wave then run forward then pick something up"*
58	2	181 f	T walk, R clap @71 (-2.22,1.35,1.14), T walk, R pick @162 (-1.42,0.15,3.68) *"a person walk forward then clap hands then walk forward then pick something up"*
59	2	184 f	T walk, R punch @71 (0.02,1.3,2.81), T walk, R jump @163 (0.34,1.05,-0.44) *"a person walk forward then punch then walk forward then jump forward"*
60	2	164 f	T run, R jump @48 (2.45,1.05,1.22), T hop, R jump @149 (0.86,1.05,3.43) *"a person run forward then jump forward then hop forward then jump forward"*
61	2	129 f	T run, R push @47 (2.33,1.2,-1.74), T run, R jump @112 (3.87,1.05,-3.38) *"a person run forward then push forward then run forward then jump forward"*
62	2	180 f	T walk, R stomp @70 (-0.44,0.1,-2.49), T walk, R squat @160 (-2.53,0.55,-1.28) *"a person walk forward then stomp then walk forward then squat down"*
63	2	182 f	T walk, R reachup @70 (-2.34,2.05,-0.71), T walk, R squat @160 (0.05,0.55,-0.94) *"a person walk forward then reach up high then walk forward then squat down"*
64	2	147 f	T walk, R throw @69 (-1.88,1.6,1.97), T run, R clap @131 (-4.1,1.35,1.16) *"a person walk forward then throw a ball then run forward then clap hands"*
65	2	186 f	T walk, R clap @77 (0.89,1.35,2.49), T walk, R clap @168 (2.91,1.35,3.57) *"a person walk forward then clap hands then walk forward then clap hands"*
66	2	168 f	T run, R kick @51 (2.64,0.55,0.33), T walk, R clap @148 (0.54,1.35,-1.78) *"a person run forward then kick then walk forward then clap hands"*
67	2	181 f	T hop, R jump @86 (1.5,1.05,-2.23), T run, R kneel @159 (2.09,0.1,0.51) *"a person hop forward then jump forward then run forward then kneel down"*
68	2	178 f	T walk, R bow @72 (-1.27,0.95,-2.27), T walk, R kneel @161 (-1.12,0.1,-4.43) *"a person walk forward then bow then walk forward then kneel down"*
69	2	179 f	T walk, R stomp @71 (-2.47,0.1,0.11), T walk, R reachup @160 (-3.03,2.05,2.35) *"a person walk forward then stomp then walk forward then reach up high"*
70	2	125 f	T run, R throw @47 (1.68,1.6,2.08), T run, R bow @110 (-0.84,0.95,3.3) *"a person run forward then throw a ball then run forward then bow"*
71	2	178 f	T walk, R kick @70 (0.71,0.55,-2.72), T walk, R bow @158 (-2.05,0.95,-2.53) *"a person walk forward then kick then walk forward then bow"*
72	2	157 f	T walk, R jump @69 (-2.12,1.05,-1.87), T run, R push @135 (0.83,1.2,-0.39) *"a person walk forward then jump forward then run forward then push forward"*
73	2	177 f	T walk, R bow @72 (-2.37,0.95,0.95), T walk, R push @161 (-4.09,1.2,-1.14) *"a person walk forward then bow then walk forward then push forward"*
74	2	163 f	T run, R stomp @51 (-0.21,0.1,2.42), T walk, R kick @145 (1.1,0.55,4.74) *"a person run forward then stomp then walk forward then kick"*

Table 17 continued

Spec	Tier	Len.	Ordered events (transition; reach @ core, anchor)
75	2	141 f	T run, R kneel @50 (2.04,0.1,1.26), T run, R kick @121 (1.52,0.55,-1.59) *"a person run forward then kneel down then run forward then kick"*
76	2	174 f	T walk, R squat @70 (-0.19,0.55,-2.28), T walk, R punch @157 (1.24,1.3,-4.71) *"a person walk forward then squat down then walk forward then punch"*
77	2	158 f	T walk, R jump @70 (-2.1,1.05,1.83), T run, R throw @137 (0.79,1.6,0.54) *"a person walk forward then jump forward then run forward then throw a ball"*
78	2	147 f	T run, R bow @46 (0.67,0.95,2.62), T walk, R throw @132 (-2.11,1.6,2.38) *"a person run forward then bow then walk forward then throw a ball"*
79	2	189 f	T hop, R stomp @86 (2.51,0.1,0.56), T walk, R wave @172 (4.69,1.55,0.98) *"a person hop forward then stomp then walk forward then wave"*
80	3	189 f	T run, R kick @46 (1.34,0.55,-2.66), T run, R wave @111 (-0.85,1.55,-3.57), T run, R pick @174 (0.17,0.15,-6.47) *"a person run forward then kick then run forward then wave then run forward then pick something up"*
81	3	190 f	T run, R bow @46 (2.61,0.95,0.31), T run, R wave @107 (1.89,1.55,2.58), T run, R throw @171 (2.56,1.6,5.19) *"a person run forward then bow then run forward then wave then run forward then throw a ball"*
82	3	189 f	T run, R jump @47 (-2.67,1.05,0.96), T run, R clap @109 (-1.2,1.35,-1.48), T run, R kick @171 (-4.06,0.55,-1.22) *"a person run forward then jump forward then run forward then clap hands then run forward then kick"*
83	3	187 f	T run, R wave @45 (-2.16,1.55,1.36), T run, R pick @106 (-0.16,0.15,-0.62), T run, R punch @169 (1.94,1.3,-2.05) *"a person run forward then wave then run forward then pick something up then run forward then punch"*
84	3	190 f	T run, R reachup @47 (2.58,2.05,-0.71), T run, R clap @109 (5.1,1.35,0.34), T run, R kneel @172 (2.85,0.1,1.55) *"a person run forward then reach up high then run forward then clap hands then run forward then kneel down"*
85	3	187 f	T run, R throw @45 (0.29,1.6,-2.65), T run, R stomp @105 (-0.52,0.1,-4.92), T run, R punch @169 (-1.65,1.3,-7.44) *"a person run forward then throw a ball then run forward then stomp then run forward then punch"*
86	3	184 f	T run, R wave @46 (-1.63,1.55,-1.76), T run, R push @108 (-2.89,1.2,-4.21), T run, R jump @168 (-1.48,1.05,-6.25) *"a person run forward then wave then run forward then push forward then run forward then jump forward"*
87	3	186 f	T run, R jump @46 (1.58,1.05,2.47), T run, R push @108 (3.87,1.2,0.28), T run, R wave @170 (2.76,1.55,-1.67) *"a person run forward then jump forward then run forward then push forward then run forward then wave"*
88	3	187 f	T run, R reachup @46 (0.9,2.05,-2.42), T run, R kneel @110 (2.91,0.1,-3.84), T run, R reachup @172 (1.36,2.05,-5.56) *"a person run forward then reach up high then run forward then kneel down then run forward then reach up high"*
89	3	183 f	T run, R push @45 (2.4,1.2,1.85), T run, R jump @106 (0.98,1.05,4.25), T run, R pick @168 (3.94,0.15,3.99) *"a person run forward then push forward then run forward then jump forward then run forward then pick something up"*
90	3	190 f	T run, R punch @46 (-2.98,1.3,-0.28), T run, R clap @107 (-1.38,1.35,-2.6), T run, R clap @171 (0.48,1.35,-4.33) *"a person run forward then punch then run forward then clap hands then run forward then clap hands"*
91	3	187 f	T run, R clap @45 (2.62,1.35,1.08), T run, R push @107 (-0.63,1.2,0.95), T run, R bow @171 (-1.74,0.95,2.95) *"a person run forward then clap hands then run forward then push forward then run forward then bow"*
92	3	189 f	T run, R bow @45 (-2.44,0.95,-1.23), T run, R bow @108 (-2.24,0.95,-3.73), T run, R squat @172 (-2.55,0.55,-1.19) *"a person run forward then bow then run forward then bow then run forward then squat down"*

Table 17 continued

Spec	Tier	Len.	Ordered events (transition; reach @ core, anchor)
93	3	188 f	T run, R throw @46 (-2.9,1.6,0.01), T run, R squat @110 (-4.72,0.55,1.36), T run, R jump @173 (-2.58,1.05,3.35) *"a person run forward then throw a ball then run forward then squat down then run forward then jump forward"*
94	3	189 f	T run, R pick @48 (-1.99,0.15,1.8), T run, R kick @111 (-4.31,0.55,-0.02), T run, R squat @173 (-4.58,0.55,-2.1) *"a person run forward then pick something up then run forward then kick then run forward then squat down"*
95	3	186 f	T run, R punch @45 (-1.91,1.3,2.17), T run, R kneel @107 (-3.07,0.1,-0.49), T run, R clap @170 (-0.89,1.35,1.38) *"a person run forward then punch then run forward then kneel down then run forward then clap hands"*
96	3	189 f	T run, R pick @46 (2.73,0.15,0.2), T run, R jump @110 (0.67,1.05,-2.2), T run, R wave @174 (3.44,1.55,-1.62) *"a person run forward then pick something up then run forward then jump forward then run forward then wave"*
97	3	190 f	T run, R clap @48 (2.16,1.35,-1.67), T run, R kick @112 (3.32,0.55,-4.11), T run, R wave @174 (1.12,1.55,-5.16) *"a person run forward then clap hands then run forward then kick then run forward then wave"*
98	3	188 f	T run, R stomp @45 (1.96,0.1,-1.91), T run, R kneel @108 (3.89,0.1,-3.35), T run, R pick @171 (6.57,0.15,-2.89) *"a person run forward then stomp then run forward then kneel down then run forward then pick something up"*
99	3	190 f	T run, R wave @48 (-1.04,1.55,2.33), T run, R squat @113 (-1.41,0.55,-0.09), T run, R bow @175 (-3.42,0.95,1.99) *"a person run forward then wave then run forward then squat down then run forward then bow"*
100	3	189 f	T run, R throw @48 (0.83,1.6,2.45), T run, R reachup @112 (3.02,2.05,1.82), T run, R punch @173 (0.89,1.3,4.18) *"a person run forward then throw a ball then run forward then reach up high then run forward then punch"*
101	3	190 f	T run, R jump @48 (1.51,1.05,-2.22), T run, R squat @113 (3.73,0.55,-2.21), T run, R clap @175 (3.17,1.35,0.47) *"a person run forward then jump forward then run forward then squat down then run forward then clap hands"*
102	3	190 f	T run, R kick @45 (-1.06,0.55,-2.43), T run, R wave @110 (1.17,1.55,-3.17), T run, R bow @175 (-0.81,0.95,-4.54) *"a person run forward then kick then run forward then wave then run forward then bow"*
103	3	187 f	T run, R kick @46 (-0.88,0.55,2.54), T run, R punch @110 (-3.42,1.3,3.55), T run, R throw @172 (-4.15,1.6,5.94) *"a person run forward then kick then run forward then punch then run forward then throw a ball"*
104	3	188 f	T run, R wave @46 (0.78,1.55,-2.43), T run, R punch @110 (-0.54,1.3,-4.74), T run, R push @173 (0.16,1.2,-1.18) *"a person run forward then wave then run forward then punch then run forward then push forward"*
105	3	184 f	T run, R bow @45 (-1.51,0.95,2.17), T run, R throw @106 (1.17,1.6,0.97), T run, R kick @168 (3.41,0.55,0.44) *"a person run forward then bow then run forward then throw a ball then run forward then kick"*
106	3	189 f	T run, R kneel @45 (0.57,0.1,2.53), T run, R bow @105 (2.76,0.95,4.01), T run, R throw @170 (5.21,1.6,4.17) *"a person run forward then kneel down then run forward then bow then run forward then throw a ball"*
107	3	188 f	T run, R reachup @45 (2.25,2.05,0.96), T run, R push @107 (4.8,1.2,-0.13), T run, R wave @171 (4.39,1.55,2.56) *"a person run forward then reach up high then run forward then push forward then run forward then wave"*
108	3	190 f	T run, R jump @45 (2.8,1.05,0.52), T run, R clap @106 (-0.34,1.35,0.52), T run, R kneel @170 (-1.25,0.1,-1.94)

Table 17 continued

Spec	Tier	Len.	Ordered events (transition; reach @ core, anchor)
			"a person run forward then jump forward then run forward then clap hands then run forward then kneel down"
109	3	187 f	T run, R stomp @47 (1.71,0.1,-1.75), T run, R punch @110 (2.18,1.3,-4.62), T run, R throw @171 (0.98,1.6,-6.76) *"a person run forward then stomp then run forward then punch then run forward then throw a ball"*
110	3	187 f	T run, R squat @47 (-1.84,0.55,1.61), T run, R punch @110 (1.09,1.3,2.04), T run, R bow @171 (-1.75,0.95,0.3) *"a person run forward then squat down then run forward then punch then run forward then bow"*
111	3	189 f	T run, R reachup @47 (1.87,2.05,-1.41), T run, R kick @110 (1.54,0.55,1.39), T run, R clap @172 (-1.09,1.35,0.77) *"a person run forward then reach up high then run forward then kick then run forward then clap hands"*
112	3	187 f	T run, R jump @45 (1.99,1.05,-1.89), T run, R kneel @107 (0.44,0.1,0.51), T run, R reachup @170 (2.76,2.05,-0.95) *"a person run forward then jump forward then run forward then kneel down then run forward then reach up high"*
113	3	190 f	T run, R clap @46 (1.49,1.35,2.13), T run, R wave @108 (2.79,1.55,-0.04), T run, R kneel @172 (3.26,0.1,2.83) *"a person run forward then clap hands then run forward then wave then run forward then kneel down"*
114	3	185 f	T run, R squat @47 (2.06,0.55,1.5), T run, R stomp @109 (3.98,0.1,3.01), T run, R punch @170 (5.21,1.3,0.23) *"a person run forward then squat down then run forward then stomp then run forward then punch"*
115	3	189 f	T run, R throw @47 (2.48,1.6,0.94), T run, R reachup @110 (1.36,2.05,2.98), T run, R squat @173 (2.37,0.55,4.96) *"a person run forward then throw a ball then run forward then reach up high then run forward then squat down"*
116	3	185 f	T run, R stomp @45 (-1.51,0.1,1.93), T run, R squat @107 (-1.96,0.55,-0.44), T run, R throw @169 (-4.71,1.6,-1.24) *"a person run forward then stomp then run forward then squat down then run forward then throw a ball"*
117	3	186 f	T run, R clap @45 (2.5,1.35,0.04), T run, R stomp @108 (1.51,0.1,-2.62), T run, R reachup @171 (-0.93,2.05,-1.86) *"a person run forward then clap hands then run forward then stomp then run forward then reach up high"*
118	3	187 f	T run, R clap @45 (1.82,1.35,-1.99), T run, R squat @107 (4.1,0.55,-2.59), T run, R push @170 (1.15,1.2,-1.75) *"a person run forward then clap hands then run forward then squat down then run forward then push forward"*
119	3	188 f	T run, R push @45 (-3.03,1.2,-0.28), T run, R wave @109 (-0.1,1.55,-1.28), T run, R jump @172 (-2.13,1.05,1) *"a person run forward then push forward then run forward then wave then run forward then jump forward"*