SCAMP: SPARSE-ANCHOR CONTROL IS ONE SMALL PROJECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Text fixes what a motion is, but not where the body must be. Authoring needs sparse anchors: chosen joints, at chosen frames, at given positions. Meeting them currently costs a conditioning branch trained for the task, or hundreds of per-clip optimisation steps in the architecture's native variables. In any generator that decodes a continuous state through a frozen differentiable decoder, the anchors ask for little and leave most of the state free: a few hundred numbers against tens of thousands, so the states that meet them form a set within a few hundred dimensions of the state space itself. Every control method is a choice inside that set, and the choices differ along the directions the anchors cannot see and the motion can. Our solver, SCAMP (sparse-anchor control as a minimum-norm projection), makes the choice that moves none of them: damped Gauss–Newton in the space of the anchors, through the frozen decoder alone, training-free, one dimensionless damping constant. Every increment is a combination of the rows of the anchors' Jacobian, so the correction is orthogonal to everything they never see, and the system solved is the request's size and not the state's. Applied unchanged to seven published generators spanning diffusion, token and latent designs, in anchor error it matches or exceeds every released control method it is measured against, and closes the anchors on the hosts that ship none. Held to those rows, a correction can only take the shapes the decoder admits, so its cost is the decoder's, and a fixed solver makes that cost measurable: every generator pays in foot skating, and how much divides by decoder family. A decoder that spreads a change of state over a window keeps its mean contact-speed skating within a small multiple of the unconstrained generator's own; an analytic recovery displaces the body itself, and skating multiplies several times over. Our generator is built to that criterion: a windowed decoder and a sampler trained through the correction, so it meets its anchors within a shallow solve at sub-second cost. The decoder's temporal support is a design criterion for controllable motion generation.

1 INTRODUCTION

A foot on a mark, a hand on a door handle, a root passing through a waypoint: the moment a text-to-motion model meets an authoring task, the specification stops being a sentence and becomes a small set of Cartesian constraints on particular joints at particular frames — *sparse anchors*. Generators meet anchors today with machinery bought at real expense: a conditioning branch trained for the task, or per-clip optimisation cast in the architecture's native variables — guidance on the sample (Karunratanakul et al., 2023; Xie et al., 2024), descent on the initial noise (Karunratanakul et al., 2024), re-imposition of observed frames during denoising (Cohan et al., 2024), descent in a learned latent space (Dai et al., 2024). The machinery is out of proportion to the thing specified: a few hundred scalar constraints, set against a state of 10^4–10^5 dimensions.

The disproportion has a mathematical root. Every one of these mechanisms runs on a generator that produces motion by decoding a continuous state $\boldsymbol{E}$ — a diffusion sample, a latent, a token model's code embedding — through a frozen differentiable decoder; call it a *host*. Under that one shared assumption, anchor control is a single problem: find the state nearest the host's own output whose decoded motion meets the anchors,

$$\min_{\boldsymbol{E}} \ \tfrac{1}{2} \left\| \boldsymbol{E} - \boldsymbol{E}_0 \right\|^2 \quad \text{subject to} \quad \boldsymbol{r}(\boldsymbol{E}) = \mathbf{0}, \tag{1}$$

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.

where $\boldsymbol{E}_0$ is the host's own sample and $\boldsymbol{r}$ stacks the anchor deviations of the decoded motion.

What that problem asks is small; what it leaves open is not. Write $\boldsymbol{J}$ for the Jacobian of $\boldsymbol{r}$: one row per anchor coordinate, m of them, against one column per coordinate of the state, P of those — a few hundred against tens of thousands. The states that meet the anchors therefore form a set of dimension $P - m$, and every control method, a trained branch or a guidance path or re-imposed frames or a latent descent, is a choice inside it. The choices are not interchangeable: the anchors are blind to those directions and the motion is not. SCAMP takes the one that moves none of them. Each accepted increment is a combination of the rows of $\boldsymbol{J}$, so it is orthogonal to $\ker \boldsymbol{J}$, and the correction as a whole is orthogonal to every direction that stayed invisible to the anchors along the way — five accepted steps enter at most $5m$ of the state's directions and leave the rest where the generator put them.

Making that choice costs one linear solve. The objective of problem 1 is exactly quadratic and only the constraint is linearised, which gives a step of the residual's own dimension, taken through the decoder rather than the sampler. The solver, SCAMP, is damped Gauss–Newton with a single damping constant made dimensionless by the Jacobian's own scale, and the system it solves is $m \times m$ on every host: the request sets its size, not the state. Applied unchanged to seven published hosts it matches or exceeds each released control method in anchor error, without training and, on MDM, MaskControl and OmniControl, in a fraction of its time (§4.1).

Because the correction lives in those rows, the shapes it can take are the decoder's and so is what they cost. Every representation admits the correction, and every one pays for it in foot skating on the way. The size of the correction fails to predict the cost: the host that moves the motion farthest pays a fraction of what the host that moves it least pays. What tracks it is the decoder's temporal support, the number of frames that feel a change of one state coordinate. A decoder that reconstructs each frame from a window of state answers an anchor with a local deformation; an analytic recovery moves the body itself, either within the anchored frames or, when the root is integrated, as a velocity bias across the whole prefix. Grouped by decoder family the hosts separate without overlap, and an intervention inside a single host moves the cost in whichever direction its decoder leaves open: widening the update's temporal basis lowers it on a frame-local decoder and buys nothing on a windowed one (§4.2).

The finding motivates a windowed decoder. We keep one and build the sampler around it: a continuous-time Markov chain over motion tokens, conditioned on the anchor residual and trained through the terminal correction (§3.6). It meets its anchors at sub-second cost.

Contributions. (i) One operator for anchor control. Damped Gauss–Newton in the space of the anchors, through the frozen decoder alone, taking the increment that moves nothing the anchors cannot see and solving $m \times m$ whatever the size of the state — training-free, one dimensionless damping constant, tuned per host nowhere. Retrofitted unchanged onto seven published generators, it matches or exceeds in anchor error the released control method of every host that ships one, and closes the anchors on the two that ship none (§4.1). **(ii) The cost of a correction, read as a property of the representation.** One fixed operator makes the generator the experimental variable, and the physical cost divides by decoder family, with the decoder's temporal support a causal factor in it: the families separate without overlap in jerk and in mean contact speed (§4.2). **(iii) A generator built to that criterion.** A codebook-geometric continuous-time Markov chain over motion tokens, keeping a windowed decoder and trained through the correction itself, reaching 0.083 m anchor error at FID 0.102 in 0.50 s per clip (§4.3).

2 Related work

Control native to motion generators. GMD (Karunratanakul et al., 2023) and OmniControl (Xie et al., 2024) add a classifier-style gradient to the denoising steps of a raw-motion diffusion model; the gradient is evaluated through the decoder's map to joint positions, which is equation 4 up to a step size in its large-μ limit. DNO (Karunratanakul et al., 2024) and MotionLCM's CLT (Dai et al., 2024) instead treat the initial noise or the latent as the decision variable and run hundreds of Adam steps through the whole sampler; our correction backpropagates through the decoder alone, never through the sampler. On the trained side, CondMDI (Cohan et al., 2024) learns a keyframe-conditioned model and imputes observed frames during sampling, and MaskControl (Pinyoanuntapong et al.,

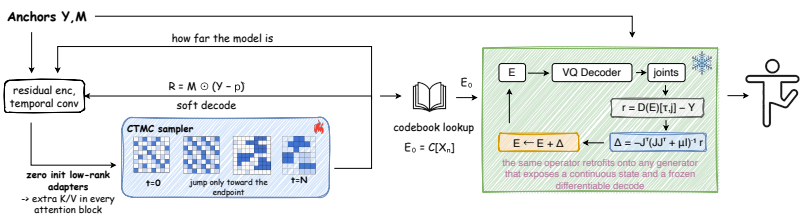

Figure 1: The anchors enter twice: during sampling as a feedback signal, the current clean posterior soft-decoded to joints $\hat{\boldsymbol{p}}$ and the model conditioned on $\boldsymbol{R} = \boldsymbol{M} \odot (\boldsymbol{Y} - \hat{\boldsymbol{p}})$ through extra keys and values; at the end as the target of one damped Gauss–Newton solve through the frozen decoder. The tokeniser is frozen throughout and its decoder is windowed, so a correction spreads over its receptive field.

2025) trains a control branch on a masked token model with token-space logit optimisation on top — the strongest prior method in our comparison. Imputation is our undamped limit: at $\mu \to 0$ with a time-selection Jacobian, whose rows are orthonormal, equation 4 is the exact projection onto the anchors — the most concentrated correction available, and §4.2 measures what that concentration costs.

Inverse problems inside a frozen prior. Outside motion this is a developed literature. RePaint (Lugmayr et al., 2022) resamples observed regions; manifold-constrained gradients (Chung et al., 2022), posterior sampling (Chung et al., 2023) and pseudo-inverse guidance (Song et al., 2023) steer each denoising step toward the measurement; the null-space decomposition (Wang et al., 2023) projects each iterate onto the measurement set — for a linear decoder, the $\mu \to 0$ limit of our own update; universal (Bansal et al., 2024) and energy (Yu et al., 2023) guidance backpropagate an arbitrary objective through the denoiser. Our operator belongs to this family, carried from gradient steps to damped Gauss–Newton steps and to a general host state. We use it as an *instrument*: one solver unchanged across hosts makes the representation the variable (§4.2). Differentiable-optimisation layers (Amos & Kolter, 2017; Pineda et al., 2022) supply the machinery for training through such a solve, as §3.6 does.

Classical constrained motion editing. Spacetime constraints (Witkin & Kass, 1988; Gleicher, 1997) and style-based IK (Grochow et al., 2004) solve the same projection, but against hand-designed regularisers rather than a learned prior. Here the regulariser is the host's own decoder, whose temporal coupling applies to every iterate, and §4.2 measures how far it reaches.

3 Method

3.1 The problem, and what a host must expose

The layer consumes two objects: a continuous state $\boldsymbol{E} \in \mathbb{R}^P$ the generator produces, and a frozen differentiable map $\mathcal{D}$ from that state to Cartesian joint positions. Any generator supplying them is a *host* — $\boldsymbol{E}$ is a diffusion sample, a latent, or a token model's continuous code embedding, and $\mathcal{D}$ composes the host's learned decoder, where it has one, with the analytic recovery of joint positions from its motion representation. On token hosts we decode the corrected state directly, so the correction reaches the motion in full rather than being rounded back onto the codebook.

A control request is a set of A anchors $\{(\tau_a, j_a, \boldsymbol{y}_a)\}$: joint j_a at position $\boldsymbol{y}_a$ in frame τ_a. Stacking the deviations gives the residual

$$\boldsymbol{r}(\boldsymbol{E}) = [\,\mathcal{D}(\boldsymbol{E})[\tau_a, j_a] - \boldsymbol{y}_a\,]_{a=1}^{A} \in \mathbb{R}^m, \qquad m = 3A, \tag{2}$$

and the request specifies problem 1: the state nearest the host's own output that zeros $\boldsymbol{r}$. Anchors are sparse, so m is orders of magnitude below P.

3.2 THE DAMPED GAUSS–NEWTON STEP

Write $\boldsymbol{J} = \partial \boldsymbol{r}/\partial \boldsymbol{E} \in \mathbb{R}^{m\times P}$. Linearising the constraint of problem 1 at $\boldsymbol{E}_0$ gives its first step: the smallest increment that meets the linearised anchors,

$$\min_{\boldsymbol{\Delta}} \ \tfrac{1}{2}\|\boldsymbol{\Delta}\|_2^2 \quad \text{subject to} \quad \boldsymbol{J}\boldsymbol{\Delta} = -\boldsymbol{r}. \tag{3}$$

Stationarity gives $\boldsymbol{\Delta} = \boldsymbol{J}^\top \boldsymbol{v}$ with $\boldsymbol{J}\boldsymbol{J}^\top \boldsymbol{v} = -\boldsymbol{r}$: the increment lies in the row space of $\boldsymbol{J}$, and the unknown is the multiplier $\boldsymbol{v}$, one coordinate per anchor coordinate. That row space is the whole of what the anchors can reach: $\boldsymbol{J}$ has m rows against P columns, so its kernel has dimension $P - m$, and the increment is orthogonal to all of it, $\langle \boldsymbol{\Delta}, \boldsymbol{n}\rangle = \boldsymbol{v}^\top \boldsymbol{J}\boldsymbol{n} = 0$ for every $\boldsymbol{n} \in \ker \boldsymbol{J}$. Confining the increment to a trust region, equivalently relaxing the equality to a quadratic penalty, adds a multiple of the identity to that system (Moré, 1978),

$$\boxed{\boldsymbol{\Delta} = -\,\boldsymbol{J}^\top \big(\boldsymbol{J}\boldsymbol{J}^\top + \mu \boldsymbol{I}_m\big)^{-1} \boldsymbol{r}} \tag{4}$$

which leaves the increment in the row space and so leaves the orthogonality where it was, and keeps the linear system at $m \times m$: the request sets its size, not the state, m=144 at the anchor density of §4 against P in the tens of thousands. The matrix inverted, $\boldsymbol{J}\boldsymbol{J}^\top + \mu \boldsymbol{I}_m$, is positive semidefinite by construction and positive definite for every $\mu > 0$, so a step exists for any request on any host. The same increment written in the state, $-(\boldsymbol{J}^\top \boldsymbol{J} + \mu \boldsymbol{I}_P)^{-1} \boldsymbol{J}^\top \boldsymbol{r}$, asks for a $P \times P$ inverse; only one of the two is computable at this size. A correction of R accepted steps therefore enters at most Rm of the state's directions and leaves the rest of the state exactly where the host put it. The iteration repeats this step from the corrected state and returns the endpoint of that path; how far it moves the decoded motion is measured per host (Displ. in Table 2). At $\mu \to 0$ and full row rank each step is the minimum-norm increment of equation 3 — the property §3.5 uses — and applied to Cartesian joint targets equation 4 is the damped least-squares step of singularity-robust inverse kinematics (Wampler, 1986; Nakamura & Hanafusa, 1986; Chiaverini et al., 1994), whose undamped min-norm form goes back to Ben-Israel (1966). What differs from that setting is the shape of $\boldsymbol{J}$. In inverse kinematics it is analytic and the unknowns are a few dozen joint angles, so the system can be square or overdetermined and the damping carries it through kinematic singularities. Here it is read from the frozen decoder by m reverse-mode products, and sparse anchors against a generator's state make $m \ll P$ by construction: the system is always the underdetermined one, and the damping is the trust region the linearisation asks for.

The loop. We iterate: build $\boldsymbol{J}$, solve equation 4, and accept the trial only if it strictly decreases the merit $\Phi(\boldsymbol{E}) = \|\boldsymbol{r}(\boldsymbol{E})\|_2^2$. A rejected trial raises the damping and re-solves the same system; acceptance lowers it. When every trial at a state is rejected, the solve returns the last accepted state (constants in Appendix A). As μ grows the step approaches $-\boldsymbol{J}^\top \boldsymbol{r}/\mu$, a descent direction for Φ, so a large enough damping yields an accepted step at any non-stationary state.

Solve depth — the number of accepted steps — is chosen per request at inference and sets the precision.

3.3 SCALE-NORMALISED DAMPING

One dimensionless constant has to serve hosts whose states differ in scale by orders of magnitude: a code embedding, a VAE latent and a normalised motion feature give Gram matrices whose diagonals are not comparable. We set

$$\mu \ = \ \lambda \cdot \frac{1}{m} \operatorname{tr}\big(\boldsymbol{J}\boldsymbol{J}^\top\big)\,, \tag{5}$$

which makes λ dimensionless: under $\boldsymbol{E} \mapsto \alpha \boldsymbol{E}$ we have $\boldsymbol{J} \mapsto \boldsymbol{J}/\alpha$, so $\mu \mapsto \mu/\alpha^2$ and $\boldsymbol{\Delta} \mapsto \alpha \boldsymbol{\Delta}$ — the iterate is exactly equivariant, and rescaling $\boldsymbol{r}$ leaves it unchanged. A single dimensionless $\lambda_0 = 10^{-2}$, set once, initialises the loop on every host in this paper.

Scalar damping is Levenberg's (Levenberg, 1944); the accept-and-retry schedule and the curvature-derived scale are Marquardt's (Marquardt, 1963), whose scaling is per-coordinate where equation 5 is a single scalar; the trust-region reading and adaptive scaling are Moré (1978)'s.

3.4 METRIC AND REPARAMETERISATION

Solving equation 3 for a reparameterised increment $\boldsymbol{\Delta} = \boldsymbol{B}\boldsymbol{\alpha}$ of smallest $\|\boldsymbol{\alpha}\|$ gives $\boldsymbol{\Delta} = -\boldsymbol{\Sigma}\boldsymbol{J}^\top(\boldsymbol{J}\boldsymbol{\Sigma}\boldsymbol{J}^\top + \mu\boldsymbol{I})^{-1}\boldsymbol{r}$ with $\boldsymbol{\Sigma} = \boldsymbol{B}\boldsymbol{B}^\top \succeq 0$, a preconditioner: a rectangular $\boldsymbol{B}$ confines the increment to the subspace its columns span. The damping scale follows it, $\mu = \lambda \operatorname{tr}(\boldsymbol{J}\boldsymbol{\Sigma}\boldsymbol{J}^\top)/m$, so a change of $\boldsymbol{B}$ leaves the effective regularisation where it was. We take $\boldsymbol{B} = \boldsymbol{I}$, and so $\boldsymbol{\Sigma} = \boldsymbol{I}$, on the evidence of Appendix B, except in the intervention of §4.2, where the columns of $\boldsymbol{B}$ are temporally smooth atoms of controlled width and widening them restricts the increment to smoother shapes.

3.5 WHAT THE STATE COSTS: TEMPORAL SUPPORT

For every damping the increment lies in the row space of $\boldsymbol{J}$. Write $\boldsymbol{J} = \boldsymbol{S}\boldsymbol{G}$ for the decoder's full Jacobian $\boldsymbol{G}$ and the selection $\boldsymbol{S}$ that reads the anchored coordinates: the decoded displacement is then a combination of the columns of $\boldsymbol{G}\boldsymbol{G}^\top\boldsymbol{S}^\top$, rows of the decoder's own Gram, which for a convolutional decoder is the autocorrelation of its temporal response and so reaches about twice as far. The host fixes the family of shapes a correction can take; the request selects which of them, and how much.

This width is a property of the frozen map $\mathcal{D}$ alone, and is measured without a solve: perturb one coordinate of the state at one frame, decode, and read how far in time the displacement extends. We report it as $w_{\text{eff}} = \sum_t d_t / \max_t d_t$ for d_t the joint displacement at frame t: one for a strictly frame-local decode, growing with the spread. Coordinates are drawn uniformly and the median reported, so it reports the reach of a typical coordinate; the few integrated root channels of an analytic recovery reach further, and §4.2 reads them in the displacement they produce.

The families differ in kind. A learned decoder of temporal receptive field w answers an anchor with a deformation supported on a window — a shape the prior already produces, since the decoder was fitted to plausible motion. An analytic recovery has no such window. From absolute-root features it is frame-local in every channel, so the increment stays inside the anchored frames. From relative-root features the root is integrated, so the row of an anchor at frame τ is constant over the whole preceding interval and the minimum-norm increment is a velocity bias of size $\|\boldsymbol{r}\|/\tau$ applied to every frame up to τ: the body is displaced, gently but continuously, for the whole prefix.

The cost is measured as foot skating, the standard physical-plausibility artefact of motion synthesis: a foot within $5\,\text{cm}$ of the ground that moves horizontally faster than $0.5\,\text{m/s}$, over a five-frame window at $20\,\text{fps}$, counts against the clip (Xie et al., 2024). Two things decide the count, the speed each frame carries and how many contact frames carry it, and the three correction shapes sit differently on both. A windowed deformation is compact and spread over the decoder's window, so the speed it adds at any one contact frame is small. A frame-local correction puts a whole displacement between two frames, far above the threshold. A root bias is small per frame and applies to the entire body, so every foot planted anywhere in the prefix is dragged with it.

3.6 A GENERATOR BUILT TO THE CRITERION

We address the shape by keeping a frozen VQ tokeniser of the MoMask family (Guo et al., 2024), whose convolutional decoder is a windowed decode, so a correction is spread over its receptive field; ours uses a single quantiser at unit temporal stride, so one token corresponds to one frame. We address the magnitude by conditioning the sampler on the anchor residual and training it *through* the terminal solve. The sampler is a continuous-time Markov chain over the token sequence, whose rates admit only jumps toward the sampled endpoint: an intermediate sequence stays on the codebook's local geometry and decodes to a nearby motion, so the anchor residual is informative from the first steps and the training gradient reaches the emitted state through it.

Codebook-geometric CTMC. We formulate generation as continuous-time discrete diffusion over the token sequence (Austin et al., 2021; Campbell et al., 2022) and instantiate its endpoint-conditioned path (Campbell et al., 2024; Gat et al., 2024) with the geometry of the frozen token embedding table. We adopt the prototype distance of Fang et al. (2026): for normalised prototypes $\bar{\boldsymbol{c}}_v$, $\delta(i,j) = \|\bar{\boldsymbol{c}}_i - \bar{\boldsymbol{c}}_j\|_2^{2p}$, whose exponent p sets how sharply the rates concentrate on near neighbours. With an increasing schedule $\beta(t) = c_\beta(t/(1-t))^\alpha$, $c_\beta, \alpha > 0$, the endpoint-conditioned

path

$$q_t(k \mid y) \;\propto\; \exp\bigl[-\beta(t)\,\delta(k,y)\bigr] \tag{6}$$

starts uniform and, for pairwise distinct prototypes, concentrates on the endpoint y as $t \to 1$. The off-diagonal rates

$$Q_t^y(i,k) \;=\; q_t(k \mid y)\,\dot\beta(t)\,\bigl[\delta(i,y) - \delta(k,y)\bigr]_+, \qquad k \neq i, \tag{7}$$

admit only jumps to codes strictly closer to y, so a position is absorbed once it reaches its endpoint and is released only when the endpoint sample changes.

Proposition 1 (The generator realises the path)**.** *For every endpoint y, the rates of equation 7 satisfy the Kolmogorov forward equation of equation 6, and their positionwise product realises the corresponding product path over sequences.*

Proposition 1 is what makes the training corruption and the sampling dynamics the same process: the prior is trained by corrupting with $q_t(\cdot \mid z^\star)$ and predicting $z^\star$, and the sampler then integrates the generator that path induces.

A masked transformer predicts, at each grid time, a clean-token posterior per position, with classifier-free guidance (Ho & Salimans, 2022) and top-k truncation; averaging equation 7 under this posterior gives the controlled rate, which we realise with one hard endpoint sample per position. Freezing the sampled rates over the interval h_n and taking the first event gives the parallel kernel

$$P_{n,\ell}^{\mathrm{next}}(k) = \begin{cases} \exp[-h_n H_{n,\ell}], & k = i, \\ \bigl(1 - \exp[-h_n H_{n,\ell}]\bigr)\, u_{n,\ell}(k)/H_{n,\ell}, & k \neq i, \end{cases} \tag{8}$$

with $u_{n,\ell}(k)$ the sampled rates out of the current token i, $H_{n,\ell}$ their sum, and all mass on i when $H_{n,\ell} = 0$ — the absorbed case.

Proposition 2 (Tau-leap consistency)**.** *The product of equation 8 over positions agrees with the posterior-averaged joint generator to first order in h_n, with remainder bounded by $Ch_n^2\,\mathbb{E}[\mathcal{H}_n^2]$ for $\mathcal{H}_n = \sum_\ell H_{n,\ell}$.*

The generator the bound refers to is the controlled one, whose clean posterior guidance and truncation tilt. Its remainder grows with the sequence length and with $\dot\beta$, which rises towards $t{=}1$, so the kernel behaves as a small-step integrator through the interior of the grid and as a commitment to the sampled endpoint at the last steps. Proofs are in Appendix G.

Anchor conditioning. The model sees anchors as a state-dependent residual: at each step the current clean posterior is soft-decoded to joint positions $\hat{\boldsymbol{p}}$, and the masked residual $\boldsymbol{R}_{\mathcal{A}} = \boldsymbol{M} \odot (\boldsymbol{Y} - \hat{\boldsymbol{p}})$ — targets $\boldsymbol{Y}$, binary observation mask $\boldsymbol{M}$ — is concatenated per frame with $\boldsymbol{Y}$ and $\boldsymbol{M}$, encoded by a temporal convolutional network, and injected through zero-initialised low-rank adapters as additional keys and values in every attention block. The conditioning is a feedback signal on the error of the model's own current clean prediction.

Solver-coupled training. The VQ encoder, decoder, codebook and text encoder stay frozen throughout. Training runs in three stages: a text-conditioned prior trained with clean-token cross-entropy over the path of equation 6; control fine-tuning under the conditioning above, over a ladder of anchor densities with first and last frames always observed; and a short on-policy calibration in which the model generates with its own sampler while the terminal solve sits inside the training graph. There the sampled prefix carries no graph, the last few states stay differentiable through hard-forward transitions (Jang et al., 2017; Maddison et al., 2017), and the solved state re-enters the loss through the identity surrogate

$$\boldsymbol{E}^{\mathrm{ST}} = \boldsymbol{E}^{\mathrm{diff}} + \mathrm{sg}\bigl[\boldsymbol{E}^{\mathrm{num}} - \boldsymbol{E}^{\mathrm{diff}}\bigr], \tag{9}$$

whose forward value is the exact numerical solve and whose Jacobian is the identity. Differentiating through equation 4 at frozen $\boldsymbol{J}$ gives $\boldsymbol{I} - \boldsymbol{J}^\top(\boldsymbol{J}\boldsymbol{J}^\top + \mu\boldsymbol{I})^{-1}\boldsymbol{J}$, which tends to the orthogonal projector onto $\ker \boldsymbol{J}$, so the exact path would annihilate precisely the gradient component along the correction just applied — the component that reports how far the emitted state was from meeting the anchors.

The anchors act twice and the solver once: they condition every sampler step, and at the last one the terminal clean state is handed to equation 4 for $R{=}2$ accepted steps under the identity metric, after which the corrected continuous state is decoded as it stands and never re-quantised (Algorithm 1). What the state it emits then costs to correct is §4.3.

Table 1: Control accuracy on the seven retrofitted hosts and our own generator (HumanML3D validation, n=1024, K=8, six joints), each against its own released control method on identical clips, anchors and numerics. Depth 0 is the state each sweep starts from; solve depth is the budget dial, reported whole. MoMask and MARDM release no sparse-control method. Our generator carries the solver in its sampler, so its incumbent column is the deployed output and the sweep a second solver on top.

			SCAMP ctrl. err (m) at depth					
Host	Its control method	Its err (m)	0	1	2	3	5	8
MDM (Tevet et al., 2023)	DNO-800	0.0379	0.8920	0.1577	0.0417	0.0096	0.0001	0.0000
MaskControl	branch + token opt.	0.0053	0.4281	0.1533	0.0550	0.0193	0.0026	0.0002
OmniControl	guidance	0.1457	0.7445	0.1568	0.0474	0.0128	0.0006	0.0000
CondMDI	keyframe cond.	0.1260	0.7723	0.0772	0.0237	0.0074	0.0006	0.0001
MotionLCM	CLT-400	0.0694	0.8120	0.4293	0.2707	0.1803	0.1006	0.0664
MoMask	—	—	0.7649	0.3470	0.1437	0.0556	0.0084	0.0011
MARDM (Meng et al., 2025)	—	—	0.8499	0.2069	0.0545	0.0117	0.0003	0.0000
Ours	in-sampler solve	0.0827	0.0827	0.0374	0.0180	0.0091	0.0026	0.0006

Table 2: What the correction costs: Ctrl, Jerk and both skate ratios read at the shallowest recorded depth at which that host reaches 1 cm, HF and Displ. at depth three (Appendix F). w_{eff} is the measured temporal support, the number of frames moved by a change of one state coordinate, a property of the frozen map $\mathcal{D}$ alone. Skate is given under both measured conventions: mean foot speed during contact (Rempe et al., 2026), and the fraction of contact frames past the threshold (Xie et al., 2024). MaskControl decodes through the frozen MoMask codebook and shares its w_{eff}; its HF and Displ. were not measured; Ctrl and both skate ratios are at n=1024, jerk, HF and Displ. at n=32.

								Skate ratio	
Host	State	w_{eff}	Depth	Ctrl (m)	Jerk	HF	Displ. (m)	speed	count
CondMDI	abs. root	1.0	3	0.0074	51.9×	0.487	0.039	6.21×	1.84×
OmniControl	rel. root	1.0	5	0.0006	27.6×	0.049	0.575	5.34×	4.76×
MDM	rel. root	1.0	3	0.0096	21.3×	0.077	0.672	4.84×	5.25×
MARDM	AE latent	28.7	5	0.0003	4.6×	0.008	0.747	2.74×	4.67×
MoMask	VQ token	49.6	5	0.0084	2.0×	0.011	0.458	1.64×	2.06×
MaskControl	VQ token	49.6	5	0.0026	1.6×	—	—	1.26×	1.39×
MotionLCM	VAE latent	54.2	40	0.0073	2.4×	0.008	0.863	1.30×	1.37×
Ours	VQ token	43.1	3	0.0091	1.1×	0.008	0.098	1.12×	1.29×

4 Experiments

4.1 Control accuracy across hosts

Seven published generators expose the interface of §3.1: MDM (Tevet et al., 2023), OmniControl (Xie et al., 2024) and CondMDI (Cohan et al., 2024) on motion features, MoMask (Guo et al., 2024) and MaskControl (Pinyoanuntapong et al., 2025) on VQ token embeddings, MARDM (Meng et al., 2025) on a continuous autoencoder latent, and MotionLCM (Dai et al., 2024) on a sequence-VAE latent (Table 2). Every sweep starts from the host's text-conditioned sample with its control machinery off, depth 0 in Table 1 (MaskControl from its control branch, Appendix F). All experiments use HumanML3D (Guo et al., 2022) with six control joints (root, head, both wrists, both feet) at K anchor frames placed uniformly including the endpoints; K=8 unless stated, giving $A = 48$ anchors and $m = 144$ residual coordinates; control error is flat out to K=32, m=576 (Appendix C). Targets come from the held-out clip; control error is the mean anchor-to-decoded-joint distance. A conformance gate checks clips, anchors, solver parameters and numerics across hosts (Appendix F).

In anchor error the sweep dominates or matches the incumbent on every host that releases a control method, and closes the anchors on the two that do not (Table 1). MotionLCM needs the most depth, the one host where the disproportion of §3.1 thins (Appendix E). Wall-clock at the depth that matches the incumbent, generation included: MDM 2.6 s against DNO's 172.3 s, MaskControl 2.0 against

Table 3: HumanML3D test split, six-point control. The upper block is uncontrolled generation, where every system performs the same text-to-motion task. The lower block is the same six-point anchor protocol throughout, so generation quality is read under a fixed control task. Control accuracy is Table 1 (validation split). Foot skate is the fraction of contact frames past the threshold (Xie et al., 2024). →: closer to real is better.

Method	FID ↓	Top-3 ↑	Diversity →	Foot skate ↓
Uncontrolled generation				
Real	0.002	0.797	9.503	—
MoMask (Guo et al., 2024)	0.082	0.797	—	—
MOGO (Fu et al., 2026)	0.064	0.801	—	—
Motion Anything (Zhang et al., 2025)	0.028	0.829	—	—
MoGeFlow (Fang et al., 2026)	0.058	0.873	9.981	—
CTMC prior (ours)	0.043	0.827	9.669	—
Six-point sparse control				
CondMDI (Cohan et al., 2024)	0.498	0.631	9.013	0.099
OmniControl (Xie et al., 2024)	0.689	0.689	9.381	0.091
TLControl (Wan et al., 2024)	4.637	0.473	8.078	0.058
MotionLCM (Dai et al., 2024)	1.013	0.676	8.722	0.141
MaskControl (Pinyoanuntapong et al., 2025)	0.116	0.842	8.940	0.082
SCAMP (ours) at 0.083 m	0.102	0.796	9.598	0.057
SCAMP (ours) at 0.0026 m	0.102	0.799	9.038	0.083

9.9, OmniControl 40.9 against 81.4, CondMDI 36.9 against 35.4, MotionLCM 11.8 against CLT-400's 13.1; MoMask 1.9 and MARDM 14.0 to a centimetre. At matched anchor error the quality cost follows the decoder family as the physical cost does; our generator closes its anchors at FID 0.078 in 0.50 s, below MaskControl's and MotionLCM's released methods (Appendix F, Table 7).

The last row is our own generator under the same operator: it emits 0.083 m and three accepted steps take it to 0.0091 m. One damping constant covers all eight, which rests on the scale normalisation of §3.3 (Appendix A).

4.2 Where the physical cost comes from

Jerk at the anchored frames measures the mechanism directly, and it separates the windowed decodes from the analytic recoveries without overlap (Table 2): 21–52× against 1.1–4.6×, and MARDM at 0.0003 m pays a fifth of what MDM pays at 0.0096 m. The measured temporal support divides them the same way, an order of magnitude apart: 1.0 frame on the analytic recoveries, 28.7 to 54.2 on the learned decoders.

CondMDI sits where §3.5 places the sharp limit: absolute-root and frame-local, its displacement confined to the anchored frames in every clip, and it pays the most. MDM and OmniControl integrate the root, so on their root channels the correction is a prefix velocity bias — a rigid drift, not a deformation — which moves the body 0.672 and 0.575 m, and they still pay 21–28×. Down-weighting those channels in the solver's metric leaves the cost where it is (Appendix B).

The size of the correction runs the other way from its cost: MotionLCM moves the motion 0.863 m and MARDM 0.747 m, the two largest displacements measured, at jerk ratios of 2.4 and 4.6, while CondMDI moves it 0.039 m at 51.9, the highest.

The downstream cost in foot skating depends on which convention measures it, and the dependence is the mechanism again. Under mean foot speed during contact the families separate as jerk does, 4.84–6.21× against 1.12–2.74×; under the fraction of contact frames above the threshold they do not, CondMDI falling to 1.84× and MARDM rising to 4.67×: a correction confined to the anchored frames makes a few very fast slides, which a mean registers and a count barely does, and one across the clip makes many slow ones.

At matched accuracy our generator closes its anchors to 0.0091 m and pays 1.12× at w_{eff}=43.1 against its deployed output and 1.37× against its raw sampler output; the separation holds on absolutes as on ratios (Appendix F).

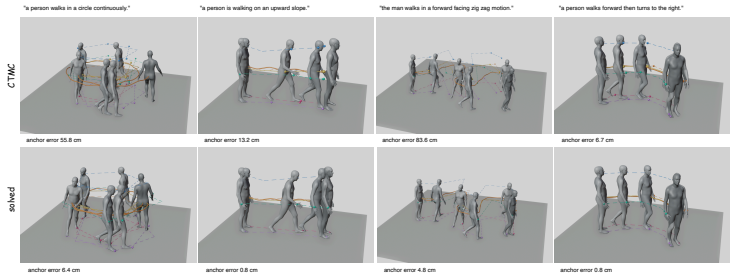

Figure 2: Four requests under the six-point protocol, $K{=}8$. Spheres are the anchor targets, one colour per joint, drawn at the frames they constrain; the bodies are the generated motion at those same frames. Above, the sample the sampler emits; below, the same sample after the terminal solve, which carries the body onto the anchors from as far as 0.84 m.

Reparameterising the update through a temporal basis of width w tests the same variable causally, at fixed weights, decoder, anchors and solver (Table 6, Appendix D). Where the decoder is frame-local the basis supplies that support from outside and the physical cost falls monotonically with it, skating 3.87 to $3.15\times$ on MDM and 5.40 to $4.30\times$ on OmniControl from $w{=}1$ to 8, read at the first depth under a centimetre ($n{=}128$); at $w{=}16$ neither host gets there. On MoMask, whose decoder already carries $w_{\text{eff}}{=}49.6$, the basis buys nothing: the high-frequency share stays at 0.009–0.011, skating at a centimetre at 1.6–$1.8\times$ over $w{=}1$ to 4, and accuracy falls, $2.1\times$ and $3.4\times$ at depth three for $w{=}8$ and 16, neither reaching a centimetre. An explicit basis supplies a smoothness the windowed decoder already has, and pays for it in the directions the solve had.

4.3 Our generator

Deployed results. The deployed configuration is Algorithm 1: $N{=}20$ sampler steps, one terminal solve, $R{=}2$ accepted steps. On the HumanML3D test split at $K{=}8$ it reaches 0.083 m control error at FID 0.102, Top-3 0.796 and diversity 9.60; latency is 0.50 s per clip unbatched (validation, $n{=}64$); three further accepted steps take it to 0.0091 m in 1.5 s and five to 0.0026 m in 2.1 s. Under the six-point protocol (Table 3) that is the best FID in the block, ahead of MaskControl's 0.116, with the diversity closest to real motion of any row and foot skating 0.057 against 0.082. Solved to MaskControl's own accuracy, 0.0026 against 0.0053 m, FID holds at 0.102 against 0.116, skating meets theirs at 0.083 against 0.082, and Top-3 is 0.799 against 0.842. The upper block places the generator before the solve: the prior scores FID 0.043 at Top-3 0.827.

Figure 2 shows what those two steps do: the sampler places the body 0.067–0.836 m from the anchors and the solve brings it to 0.008–0.064 m, closing the largest gap by $17\times$.

Trained with $R{=}1$ instead of $R{=}2$, the same recipe deploys to 0.166 and 0.087 m at the two inference depths against the deployed checkpoint's 0.165 and 0.083 (validation, $n{=}1024$): the second step buys what a step of the solver buys. Trained post hoc instead, with no solver in its sampler, the same model emits 0.367 m against 0.314 and stands at 0.107 m after the same two steps (Appendix H).

5 Conclusion

Sparse anchors fix a few hundred directions of a state with tens of thousands; what they leave open is the rest. A training-free damped Gauss–Newton solve of the anchor residual, one dimensionless damping constant and no per-host tuning, takes the increment that moves none of it, and in anchor error it matches or exceeds every released control method it is measured against.

What a correction may then look like is the decoder's to decide, and holding the solver fixed measures what its shapes cost: at matched anchor accuracy the decoder families separate without overlap in jerk and in mean contact speed. Built to that criterion, our generator reaches 0.083 m at FID 0.102. Temporal support is measurable in advance on any host that exposes a differentiable decode, and is a design choice.

ETHICS STATEMENT

This work synthesises human motion from text and Cartesian anchors on public HumanML3D data, and releases no new human data. The generators it retrofits are public released checkpoints, used under their own licences.

REPRODUCIBILITY STATEMENT

The solver is equation 4 with the schedule and constants of Appendix A; the protocol, conformance gate and metric conventions are in Appendix F; the generator's three training stages are in Appendix H. Code, the gate's output and the per-sample records will accompany the paper.

STATEMENT ON LARGE LANGUAGE MODEL USE

Large language models were used to draft and edit prose and to write plotting and bookkeeping scripts. They were not used to design the method, run experiments, or produce any number reported here; every figure in the paper is traced to a recorded run.

REFERENCES

Brandon Amos and J. Zico Kolter. OptNet: Differentiable optimization as a layer in neural networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 136–145. PMLR, 2017. URL http://proceedings.mlr.press/v70/amos17a.html.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete State-Spaces. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 17981–17993, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/958c530554f78bcd8e97125b70e6973d-Abstract.html.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=pzpWBbnwiJ.

Adi Ben-Israel. A newton-raphson method for the solution of systems of equations. *Journal of Mathematical Analysis and Applications*, 15(2):243–252, Aug 1966. ISSN 0022-247X. doi: 10.1016/0022-247x(66)90115-6. URL http://dx.doi.org/10.1016/0022-247x(66)90115-6.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/b5b528767aa35f5b1a60fe0aaeca0563-Abstract-Conference.html.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi S. Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 5453–5512. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/campbell24a.html.

S. Chiaverini, B. Siciliano, and O. Egeland. Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator. *IEEE Transactions on Control Systems Technology*, 2(2):123–134, June 1994. ISSN 1063-6536. doi: 10.1109/87.294335. URL http://dx.doi.org/10.1109/87.294335.

Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/a48e5877c7bf86a513950ab23b360498-Abstract-Conference.html.

Hyungjin Chung, Jeongsol Kim, Michael Thompson McCann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=OnD9zGAGT0k.

Setareh Cohan, Guy Tevet, Daniele Reda, Xue Bin Peng, and Michiel van de Panne. Flexible motion in-betweening with diffusion models. In Andres Burbano, Denis Zorin, and Wojciech Jarosz (eds.), *ACM SIGGRAPH 2024 Conference Papers, SIGGRAPH 2024, Denver, CO, USA, 27 July 2024- 1 August 2024*. ACM, 2024. doi: 10.1145/3641519.3657414. URL https://doi.org/10.1145/3641519.3657414.

Wenxun Dai, Ling-Hao Chen, Jingbo Wang, Jinpeng Liu, Bo Dai, and Yansong Tang. MotionLCM: Real-time controllable motion generation via latent consistency model. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XVI*, volume 15074 of *Lecture Notes in Computer Science*, pp. 390–408. Springer, 2024. doi: 10.1007/978-3-031-72640-8_22. URL https://doi.org/10.1007/978-3-031-72640-8_22.

Pengcheng Fang, Tengjiao Sun, Xiaoyu Zhan, Xiaohao Cai, and Dongjie Fu. MoGeFlow: Flowing through motion codebook geometry for text-to-motion generation. *arXiv preprint arXiv:2606.11656*, 2026.

Dongjie Fu, Tengjiao Sun, Pengcheng Fang, Xiaohao Cai, and Hansung Kim. MOGO: Residual quantized hierarchical causal transformer for real-time and infinite-length 3D human motion generation. In *Fortieth AAAI Conference on Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 3994–4002. AAAI Press, 2026. doi: 10.1609/AAAI.V40I5.37402.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/f0d629a734b56a642701bba7bc8bb3ed-Abstract-Conference.html.

Michael Gleicher. Motion editing with spacetime constraints. In Andy van Dam (ed.), *Proceedings of the 1997 Symposium on Interactive 3D Graphics, SI3D '97, Providence, RI, USA, April 27-30, 1997*, pp. 139–148, 195. ACM, 1997. doi: 10.1145/253284.253321. URL https://doi.org/10.1145/253284.253321.

Keith Grochow, Steven L. Martin, Aaron Hertzmann, and Zoran Popovic. Style-based inverse kinematics. *ACM Trans. Graph.*, 23(3):522–531, 2004. doi: 10.1145/1015706.1015755. URL https://doi.org/10.1145/1015706.1015755.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3D human motions from text. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 5142–5151. IEEE, 2022. doi: 10.1109/CVPR52688.2022.00509. URL https://doi.org/10.1109/CVPR52688.2022.00509.

Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. MoMask: Generative masked modeling of 3D human motions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 1900–1910. IEEE, 2024. doi: 10.1109/CVPR52733.2024.00186. URL https://doi.org/10.1109/CVPR52733.2024.00186.

Jonathan Ho and Tim Salimans. Classifier-Free diffusion guidance. *CoRR*, abs/2207.12598, 2022. doi: 10.48550/ARXIV.2207.12598. URL https://doi.org/10.48550/arXiv.2207.12598.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=rkE3y85ee.

Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. Guided motion diffusion for controllable human motion synthesis. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 2151–2162. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00205. URL https://doi.org/10.1109/ICCV51070.2023.00205.

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, Seattle, WA, USA, June 16-22, 2024*, pp. 1334–1345. IEEE, 2024. doi: 10.1109/CVPR52733.2024.00133. URL https://doi.org/10.1109/CVPR52733.2024.00133.

Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168, July 1944. ISSN 0033-569X. doi: 10.1090/qam/10666. URL http://dx.doi.org/10.1090/qam/10666.

Andreas Lugmayr, Martin Danelljan, Andrés Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 11451–11461. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01117. URL https://doi.org/10.1109/CVPR52688.2022.01117.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=S1jE5L5gl.

Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, June 1963. ISSN 2168-3484. doi: 10.1137/0111030. URL http://dx.doi.org/10.1137/0111030.

Zichong Meng, Yiming Xie, Xiaogang Peng, Zeyu Han, and Huaizu Jiang. Rethinking diffusion for text-driven human motion generation: Redundant representations, evaluation, and masked autoregression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 27859–27871. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.02594.

Jorge J. Moré. *The Levenberg-Marquardt algorithm: Implementation and theory*, pp. 105–116. Springer Berlin Heidelberg, 1978. ISBN 9783540359722. doi: 10.1007/bfb0067700. URL http://dx.doi.org/10.1007/bfb0067700.

Yoshihiko Nakamura and Hideo Hanafusa. Inverse kinematic solutions with singularity robustness for robot manipulator control. *Journal of Dynamic Systems, Measurement, and Control*, 108(3):163–171, Sept 1986. ISSN 1528-9028. doi: 10.1115/1.3143764. URL http://dx.doi.org/10.1115/1.3143764.

Luis Pineda, Taosha Fan, Maurizio Monge, Shobha Venkataraman, Paloma Sodhi, Ricky T. Q. Chen, Joseph Ortiz, Daniel DeTone, Austin S. Wang, Stuart Anderson, Jing Dong, Brandon Amos, and Mustafa Mukadam. Theseus: A library for differentiable nonlinear optimization. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/185969291540b3cd86e70c51e8af5d08-Abstract-Conference.html.

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 *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pp. 9955–9965. IEEE, 2025. doi: 10.1109/ICCV51701.2025.00928. URL https://doi.org/10.1109/ICCV51701.2025.00928.

Davis Rempe, Mathis Petrovich, Ye Yuan, Haotian Zhang, Xue Bin Peng, Yifeng Jiang, Tingwu Wang, Umar Iqbal, David Minor, Michael de Ruyter, Jiefeng Li, Chen Tessler, Edy Lim, Eugene Jeong, Sam Wu, Ehsan Hassani, Michael Huang, Jin-Bey Yu, Chaeyeon Chung, Lina Song, Olivier Dionne, Jan Kautz, Simon Yuen, and Sanja Fidler. Kimodo: Scaling controllable human motion generation. *CoRR*, abs/2603.15546, 2026.

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-Guided diffusion models for inverse problems. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=9_gsMA8MRKQ.

Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit Haim Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Charles Wampler. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Transactions on Systems, Man, and Cybernetics*, 16 (1):93–101, Jan 1986. ISSN 0018-9472. doi: 10.1109/tsmc.1986.289285. URL http://dx.doi.org/10.1109/tsmc.1986.289285.

Weilin Wan, Zhiyang Dou, Taku Komura, Wenping Wang, Dinesh Jayaraman, and Lingjie Liu. TLControl: Trajectory and language control for human motion synthesis. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XXXVII*, volume 15095 of *Lecture Notes in Computer Science*, pp. 37–54. Springer, 2024. doi: 10.1007/978-3-031-72913-3_3.

Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-Shot image restoration using denoising diffusion Null-Space model. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=mRieQgMtNTQ.

Andrew P. Witkin and Michael Kass. Spacetime constraints. In Richard J. Beach (ed.), *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1988, Atlanta, Georgia, USA, August 1-5, 1988*, pp. 159–168. ACM, 1988. doi: 10.1145/54852.378507. URL https://doi.org/10.1145/54852.378507.

Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. OmniControl: Control any joint at any time for human motion generation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=gd0lAEtWso.

Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. FreeDoM: Training-Free Energy-Guided conditional diffusion model. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 23117–23127. IEEE, 2023. doi: 10.1109/ICCV51070.2023.02118. URL https://doi.org/10.1109/ICCV51070.2023.02118.

Zeyu Zhang, Yiran Wang, Wei Mao, Danning Li, Rui Zhao, Biao Wu, Zirui Song, Bohan Zhuang, Ian Reid, and Richard Hartley. Motion anything: Any to motion generation. *arXiv preprint arXiv:2503.06955*, 2025.

A SOLVER DETAILS

The loop, in full. At outer step k the solver holds a state $\boldsymbol{E}_k$, evaluates $\boldsymbol{r}_k = \boldsymbol{r}(\boldsymbol{E}_k)$ and builds $\boldsymbol{J}_k$ by m reverse-mode products through $\mathcal{D}$. A trial uses $\mu_k = \lambda_k \frac{1}{m}\operatorname{tr}(\boldsymbol{J}_k\boldsymbol{J}_k^\top)$ in equation 4, and is accepted when $\Phi(\boldsymbol{E}_k + \boldsymbol{\Delta}) < \Phi(\boldsymbol{E}_k)$ strictly, for the merit Φ of §3.2. On rejection $\lambda \leftarrow 10\lambda$ and the same linear system is re-solved with the new damping; the Jacobian is rebuilt only after an accepted step. Up to 16 trials are allowed per step on the retrofitted hosts, and four in the deployed configuration of §4.3. On acceptance $\lambda \leftarrow \max(\lambda/2, 10^{-6})$ and is carried into the next step rather than reset. The solve terminates when the mean squared residual Φ/m falls below 10^{-10}, the deployed configuration of §4.3 using 10^{-5} in its place (Appendix H); when a step exhausts its trials, when $\operatorname{tr}(\boldsymbol{J}\boldsymbol{J}^\top)$ underflows, or at the step budget; it returns the last accepted state. The $m \times m$ system is solved by LU with partial pivoting in single precision with TF32 disabled; a non-finite solution is treated as a rejected trial.

What one accepted step costs. Building $\boldsymbol{J}$ costs m reverse-mode products through the frozen decoder; forming $\boldsymbol{J}\boldsymbol{J}^\top$ costs $O(m^2P)$; the solve costs $O(m^3)$; recovering the primal step costs $O(mP)$. With $m = 144$ and $P \in [10^4, 10^5]$ the decoder passes dominate. A rejected trial adds one $O(m^3)$ re-solve and nothing else.

Reported quantities. Control error is the mean over anchors of the Euclidean distance between the decoded joint and its target, in metres; it is a different functional from the merit Φ. Relative state motion is $\|\boldsymbol{E}_R - \boldsymbol{E}_0\|_2/\|\boldsymbol{E}_0\|_2$, a latent-space quantity. Steps-to-a-centimetre is the first accepted step at which control error falls below 10^{-2} m.

Scale normalisation. Replacing equation 5 by the same *absolute* damping on every host breaks the transfer: step-3 control error (n=32) rises from 0.0435 to 0.3920 m on MoMask and from 0.0017 to 0.0166 m on MDM, while MotionLCM, whose Gram scale sits near unity, is unchanged at 0.204 m.

Gauss–Newton against first-order descent. The same state, decoder, residual and clips admit any optimiser, so we ran Adam on $\frac{1}{2}\|\boldsymbol{r}\|^2$ through the frozen decoder alone, on MDM and MoMask (validation, n=64), with the learning rate set to η times the standard deviation of the state, $\eta \in \{0.01, 0.03, 0.1\}$, 400 iterations, and the best η chosen after the fact; timing covers the update only, as for the solver, in a run separate from the timings of §4.1. On MDM, Adam stands at 0.041 m after 0.46 s and 0.016 m after 0.92 s, where the solver stands at 0.038 m (two steps, 0.66 s) and 0.010 m (three, 0.99 s); at 1.3 s the solver is below 10^{-4} m at depth five and Adam is at 0.006 m after 1.85 s. On MoMask, Adam is ahead early, 0.024 m at 0.79 s against the solver's 0.131 m at two steps, the two meet at a centimetre (0.011 m at 1.56 s against 0.011 m at depth five, 1.95 s), and below it the solver pulls away, 0.0002 m at depth sixteen (3.5 s) against 0.005 m after 400 iterations (3.1 s). Mean contact-speed skating is the same for both optimisers at matched error, 4.5–$4.8\times$ on MDM and 1.5–$1.6\times$ on MoMask: the physical cost belongs to the decoder, not to the optimiser. What the curvature buys is the last order of magnitude, from the centimetre down, at equal wall-clock.

B CHOICE OF METRIC

Replacing the identity by $\boldsymbol{\Sigma}$ in equation 3 gives $\boldsymbol{\Delta} = -\boldsymbol{\Sigma}\boldsymbol{J}^\top(\boldsymbol{J}\boldsymbol{\Sigma}\boldsymbol{J}^\top + \mu\boldsymbol{I})^{-1}\boldsymbol{r}$. We compared the identity against a diagonal state covariance and the full codebook covariance on our own generator, sweeping solve depth on the validation split (Table 4). Identity and diagonal track each other at every depth on both axes. The codebook covariance reaches the anchors more slowly and costs a factor of four in FID from the first step onward, so we use the identity throughout.

Depth zero is the untouched generation, and its FID of 0.171 is the highest in the identity column: the first accepted step reduces FID to 0.090 while cutting control error by 46%, and FID then stays below the untouched value.

The metric also fixes which coordinates a correction is cheap in, so it is the second place the shape of §3.5 could come from. On the two hosts whose increment spreads over the prefix we varied it

Table 4: Solve depth against solver metric, on the post-hoc-trained checkpoint whose sampler carries no solver, validation split, n=1504. Identity and diagonal track each other on both axes; the codebook covariance costs a factor of four in FID from the first step.

	Identity		Diagonal		Codebook cov.	
Depth	Ctrl (m)	FID	Ctrl (m)	FID	Ctrl (m)	FID
0	0.3573	0.1714	0.3573	0.1714	0.3573	0.1714
1	0.1943	**0.0901**	0.1906	**0.0888**	0.2047	0.3571
2	0.1031	0.0973	0.0994	0.0957	0.1004	0.3499
3	0.0523	0.0984	0.0486	0.0959	0.0572	0.3776
4	0.0257	0.1003	0.0240	0.0983	0.0377	0.3933
5	**0.0129**	0.1040	**0.0119**	0.1026	0.0259	0.4111

directly: $\mathbf{\Sigma} = \mathrm{diag}(\boldsymbol{w})$ with $\boldsymbol{w}$ reduced on the integrated root channels and unity elsewhere, every row read at the shallowest recorded depth reaching 1 cm, validation split, n=32. Reducing those channels tenfold moves the contact-speed ratio by under one percent — 3.89 to 3.87× on MDM, 5.43 to 5.44× on OmniControl — and jerk by under two percent, while the solve takes two further accepted steps to reach the same centimetre. The cost stays where the decoder puts it.

C Anchor density

Table 5: Anchor density on the test split. K=8 sits exactly at the shipped Jacobian row cap of 48 pairs; K=16 and K=32 use an eval-time cap of 192 so that every cell has full coverage. Match is the Guo matching score, the mean text–motion embedding distance, for which real motion scores 2.974.

K	Ctrl (m)	FID	Top-1	Top-3	Diversity	Foot skate	Match
2	0.0461	0.2294	0.4907	0.7720	9.8485	0.0438	3.1195
4	0.0832	0.1511	0.4955	0.7843	9.7391	0.0523	3.0573
8	0.0830	0.1016	0.5119	0.7955	9.5975	0.0570	3.0209
16	0.0845	0.0674	0.5112	0.7881	9.6312	0.0588	3.0054
32	0.0797	0.0580	0.5112	0.7931	9.5536	0.0558	3.0159

Control error is flat from K=4 to K=32, inside 0.079–0.085 m, while the residual dimension $m = 18K$ grows from 72 to 576: the solve holds as the constraint set grows by a factor of eight. Across the whole sweep, from K=2, Top-3 spans 0.772–0.796 and diversity 9.55–9.85.

D Temporal basis of the update

The update is confined to a temporal basis of width w frames — the increment for each state channel is the linear interpolation of $\lceil T/w \rceil$ knots placed w frames apart, the first and last on the first and last frame, so each knot carries a triangular bump of support $2w$ frames; the basis has rank $C\lceil T/w \rceil$ for C channels, carries no normalisation, and at w=1 is the identity, so that arm is the unmodified solve. On the frame-local hosts w is in motion frames; on MoMask the basis lives on the token axis of its stride-four tokeniser, so w is in tokens, $4w$ frames. Everything else is held: weights, frozen decoder, clips, anchors, damping and accept schedule. Reading the intervention against the temporal support the host's own decoder already carries gives the two directions in §4.2.

E The one host where the disproportion thins

§3.1 rests on a disproportion: a few dozen scalar constraints against a state of 10^4–10^5 dimensions, which is what makes the projection small. MotionLCM is the one host in this study where that margin closes. Its sequence-VAE latent is 512 dimensions, and the six-point request at K=8 fixes 144 of them — 28 percent, against a state of tens of thousands of coordinates on every other host.

The solve behaves as the disproportion predicts. It reaches 1 cm at depth 40, where CondMDI, MDM and our own generator reach it at depth three and OmniControl, MARDM, MoMask and MaskCon-

Table 6: Intervention on a fixed host (n=128): the update reparameterised through a temporal basis of width w (frames on MDM and OmniControl, stride-four tokens on MoMask); weights, decoder, anchors and solver unchanged. Each host is labelled with the temporal support its own decoder already carries. Ctrl is a ratio to that host's own w=1 arm, in which the basis is the identity and the solve is the unmodified one, read mid-descent at depth three; Final err is the error at the end of the sixteen-step sweep; Skate is the ratio of mean contact speed to the unconstrained generation, read at the first recorded depth at which that width's error is under a centimetre (— where the sweep never reaches it); HF is absolute, read at the end of the sweep.

		basis width w				
Host		1	2	4	8	16
MDM	Ctrl	1.00×	0.73×	0.69×	0.89×	3.09×
w_{eff}=1.0	Final err (m)	0.0000	0.0000	0.0000	0.0014	0.0251
	Skate at 1 cm	3.87×	3.71×	3.48×	3.15×	—
	HF	0.062	0.016	0.012	0.012	0.012
OmniControl	Ctrl	1.00×	0.31×	0.33×	0.66×	2.49×
w_{eff}=1.0	Final err (m)	0.0000	0.0000	0.0000	0.0003	0.0138
	Skate at 1 cm	5.40×	5.22×	4.88×	4.30×	—
	HF	0.045	0.013	0.010	0.010	0.011
MoMask	Ctrl	1.00×	0.94×	1.23×	2.13×	3.36×
w_{eff}=49.6	Final err (m)	0.0001	0.0001	0.0013	0.0111	0.0472
	Skate at 1 cm	1.67×	1.58×	1.79×	—	—
	HF	0.009	0.009	0.010	0.010	0.011

trol at depth five; the damping schedule accepts at every depth and the residual falls monotonically, 0.812 m at depth zero to 0.0664 at 8, 0.0113 at 32, 0.0073 at 40 and 0.0053 at 48. What it buys per step is simply smaller, and each Jacobian row costs a VAE decode, so depth 40 takes 30.0 s of solve against a generation that takes 0.015 s.

The host's released control method stops at 0.0694 m (Table 1), its 400 latent-optimisation steps taking 13.1 s on the timing protocol of §4.1; the solve reaches 0.0073 m, nine times nearer, in 30.0 s.

F Reproducibility

How Table 2 is read. Jerk is the ratio of jerk at the anchored frames to the unconstrained generation. HF share is the fraction of the per-frame displacement magnitude's power spectrum lying above 5 Hz, and Displ. is that magnitude's mean; both are absolute. Jerk and the contact-speed skate ratio are taken against that host's own unconstrained generation, whose absolute values are, in m/s of contact speed and in metres per frame cubed at 20 fps: CondMDI 0.251 and 0.0169, OmniControl 0.135 and 0.0035, MDM 0.163 and 0.0045, MARDM 0.249 and 0.0077, MoMask 0.196 and 0.0077, MaskControl 0.182 and 0.0070, MotionLCM 0.271 and 0.0082, ours 0.201 and 0.0102. Ours is the deployed output, already corrected by its own two accepted steps; against the state the sampler emits before that solve (Appendix H) the contact-speed ratio at depth three is 1.37× and the count ratio 2.03×. Read as absolutes at the matched depth the two groups still separate without overlap, at 0.72–1.56 against 0.23–0.68 in skate and 0.097–0.879 against 0.011–0.035 in jerk, with ours among the lowest on both. Ctrl and both skate ratios are at n=1024; jerk, HF and Displ. at n=32; w_{eff} at 16 clips with 32 probes each; on a learned decoder every state coordinate feeds every feature channel of the frames in its window, the integrated root among them, so the reach reported there includes the recovery. Every row is taken at the shallowest recorded depth reaching 1 cm, which is three for CondMDI, MDM and our generator, five for OmniControl, MARDM, MoMask and MaskControl, and forty for MotionLCM. HF and Displ. move by under three percent between depth three and depth sixteen, and are read at depth three, at sixteen for MotionLCM, the deepest depth at which they were recorded.

Quality against the released control method. Table 7 scores three hosts under one evaluator, every row, generated and reference alike, decoded to joints and its 263-d features re-extracted from them, so the evaluator sees only what the control moved: the host's unconstrained sample, its released control method, and SCAMP at the first recorded depth that matches the released method's anchor

error, with the wall-clock of §4.1. The quality cost orders by decoder family as the physical cost of Table 2 does. On the windowed host SCAMP reaches half the released method's anchor error in a fifth of its time at equal Top-3, paying FID 0.236 against 0.103; on the VAE host it matches the error at parity in time and pays more; on the frame-local analytic host the correction is the sharp limit of §3.5, and the quality goes with it. Our generator, built to the criterion and trained through the solve, closes its anchors at FID 0.078 in 0.50 s, below the released control methods of MaskControl and MotionLCM at a fraction of their time.

Table 7: Generation quality against each host's released control method (HumanML3D validation, one evaluator, features re-extracted from decoded joints for every row including the reference pool). Depth is the first recorded depth of Table 1 at or under the released method's error; time is generation plus the solve to that depth, as in §4.1. n=1024; CondMDI n=256, whose three rows are comparable with each other only. Ctrl is measured within this run; Table 1 carries the campaign's values. Contact speed in m/s.

Host	Arm	Ctrl (m)	Time (s)	FID ↓	Top-3 ↑	Div.	Contact
MaskControl (windowed)	control branch only	0.428	—	0.049	0.800	9.59	0.182
	released, branch + opt. $100/600$	0.0050	9.9	0.103	0.813	9.31	0.206
	SCAMP, depth 5	0.0026	2.0	0.236	0.806	9.18	0.230
MotionLCM (VAE)	unconstrained	0.826	—	0.256	0.833	9.25	0.271
	released: CLT-400	0.069	13.1	0.186	0.802	9.13	0.245
	SCAMP, depth 8	0.068	11.8	1.05	0.736	8.45	0.294
CondMDI (frame-local)	unconstrained (uncond. ckpt)	0.751	—	0.318	0.773	9.21	0.239
	released, keyframe ckpt	0.131	35.4	0.173	0.742	8.70	0.218
	SCAMP, depth 1	0.075	36.9	14.2	0.559	5.13	1.488
Ours (windowed, trained through solve)	deployed, R=2	0.083	0.50	0.078	0.798	9.36	0.201

One protocol for every system. All hosts receive the same clip list, the same anchor frames and the same targets. An automated conformance gate verifies byte-level equality of the clip list and the anchor tensor, together with solver parameters, damping constant, row cap, and the TF32 and seeding state, before any host is measured; its output ships with the code. Timing is per host on one card, and the released-budget DNO measurements were taken on an H100 and the rest on one A100. Clips are generated one at a time (n=64), and the sweep is one solve of sixteen accepted steps read at the depths of Table 1 (MotionLCM's continues to depth 48, Appendix E); the times of §4.1 are generation plus the solve up to the depth that matches the incumbent (MDM three, MaskControl five, OmniControl two, CondMDI one, MotionLCM eight; MoMask and MARDM to a centimetre, five), read from the same runs. Generation runs each host's released sampler: MDM at 100 DDIM steps; OmniControl and CondMDI with their 1000-step samplers, OmniControl at its fixed 196 frames; MoMask and MaskControl's branch at 10 masked steps, MaskControl's own method adding 100 token-optimisation iterations per step and 600 at the end; MARDM at 18; MotionLCM at one. The starting states are the hosts' text-conditioned samples with their control machinery off: OmniControl without its spatial guidance, CondMDI's unconditional checkpoint, MotionLCM before its latent optimisation; MaskControl's sweep starts from its trained control branch's output with the token optimisation switched off, since that branch is the model.

Noise floor. Twenty repetitions of the deployed configuration give 95 percent intervals of ± 0.0036 in FID, $\pm 7 \times 10^{-4}$ m in control error, $\pm 4 \times 10^{-4}$ in foot skating and below 10^{-4} in jerk. On the retrofitted hosts, one configuration run twice on one card (n=8, identical seed) gives a spread in control error of 1.8×10^{-4} m averaged over the depth grid, 1.2×10^{-3} m at depth three where the solve is mid-descent and the accept–retry schedule can take a different branch, and 9×10^{-5} m from depth eight onwards, where the residual is small and the step is essentially determined.

Metric conventions. Foot skating is computed on the clip truncated to its true length. Every skating number in this paper is computed under the truncated convention on one evaluation stack.

What the backward pass carries. The solve enters the graph through the straight-through surrogate of equation 9. Gradient flows through the last few sampler states and the differentiable suffix. Endpoint samples and suffix transitions are hard on the forward pass; the discrete row selection of

the next generator and the truncated support of the guided posterior are held fixed in the backward pass; the sampled prefix carries no graph.

G PROOFS

G.1 PROPOSITION 1

Fix an endpoint y, write $\delta_k = \delta(k, y)$ and $\bar{\delta}_t = \sum_i q_t(i \mid y)\delta_i$. Differentiating equation 6,

$$\frac{\mathrm{d}}{\mathrm{d}t} q_t(k \mid y) = \dot{\beta}(t)\, q_t(k \mid y)\, [\bar{\delta}_t - \delta_k]. \tag{10}$$

For the rates of equation 7, using $[a]_+ - [-a]_+ = a$,

$$\begin{aligned}\sum_i q_t(i \mid y)\, Q_t^y(i,k) &= \sum_{i \neq k} q_t(i \mid y)\, q_t(k \mid y)\, \dot{\beta}(t)[\delta_i - \delta_k]_+ - q_t(k \mid y) \sum_{j \neq k} q_t(j \mid y)\, \dot{\beta}(t)[\delta_k - \delta_j]_+ \\ &= \dot{\beta}(t)\, q_t(k \mid y) \sum_i q_t(i \mid y)[\delta_i - \delta_k] = \dot{\beta}(t)\, q_t(k \mid y)[\bar{\delta}_t - \delta_k],\end{aligned} \tag{11}$$

which is equation 10. For an endpoint sequence $\boldsymbol{y} = (y_1, \ldots, y_L)$ the product path $q_t(\boldsymbol{x} \mid \boldsymbol{y}) = \prod_\ell q_t(x_\ell \mid y_\ell)$ and the single-coordinate joint generator $\mathcal{Q}_t^{\boldsymbol{y}}(\boldsymbol{x}, \boldsymbol{x}^{\ell \to k}) = Q_t^{y_\ell}(x_\ell, k)$, with zero rate between sequences differing in more than one coordinate, satisfy the corresponding forward equation: the time derivative of the product is a sum of L terms, each differentiating one factor, and the joint generator moves probability only along single-coordinate replacements, so the single-coordinate identity applies termwise. □

G.2 PROPOSITION 2

Condition on the endpoint samples, so all rates are fixed. For one position with current token i and exit rate $H_{n,\ell}$, the bounds $|e^{-x} - 1 + x| \le x^2/2$ and $|1 - e^{-x} - x| \le x^2/2$ for $x \ge 0$ give

$$P_{n,\ell}^{\mathrm{next}} = \delta_i + h_n\, Q_\ell(i, \cdot) + r_\ell, \qquad \|r_\ell\|_{\mathrm{TV}} \le (h_n H_{n,\ell})^2, \tag{12}$$

where $Q_\ell(i, \cdot)$ places $u_{n,\ell}(k)$ on each $k \neq i$ and $-H_{n,\ell}$ on i, so $\|h_n Q_\ell(i, \cdot)\|_{\mathrm{TV}} = 2h_n H_{n,\ell}$. Expanding $\bigotimes_\ell P_{n,\ell}^{\mathrm{next}}$ around $\bigotimes_\ell \delta_{x_\ell}$: the empty selection contributes $f(\boldsymbol{x})$; the single selections contribute h_n times the sampled joint single-coordinate generator applied to f, plus r_ℓ terms bounded by $\|f\|_\infty (h_n \mathcal{H}_n)^2$; and every term selecting at least two non-identity factors is bounded in total by $\|f\|_\infty \big(\prod_\ell (1 + a_\ell) - 1 - \sum_\ell a_\ell\big)$ with $a_\ell = 2h_n H_{n,\ell} + (h_n H_{n,\ell})^2$, which is at most an absolute constant times $\|f\|_\infty (h_n \mathcal{H}_n)^2$ when $h_n \mathcal{H}_n \le 1$, and at most $2\|f\|_\infty \le 2\|f\|_\infty (h_n \mathcal{H}_n)^2$ otherwise, since all kernels involved are probability measures. Taking the expectation over the independent per-position endpoint samples replaces the sampled rates by the posterior-averaged rates in the first-order term and gives $|R_n| \le C\, h_n^2\, \mathbb{E}[\mathcal{H}_n^2]\, \|f\|_\infty$. □

The bound depends on vocabulary size and sequence length only through $\mathcal{H}_n$, which grows with L and with $\dot{\beta}(t)$. Since $\dot{\beta}$ diverges as $t \to 1$, the first-order regime is the interior of the grid; at the final steps $h_n \mathcal{H}_n \gg 1$ and the kernel is in effect a commitment to the sampled endpoint.

H TRAINING RECIPE

Solver-coupled versus post-hoc training. Both arms are measured from the state their sampler emits before any solve, under the shared solver of Table 1 (validation, n=1024). The solver-coupled sampler emits 0.314 m against 0.367 m from the same model trained post hoc, whose sampler carries no solver. Two accepted steps take them to 0.083 m, the deployed output, and to 0.107 m; a centimetre takes 4.36 accepted steps against 4.56, moving the state 0.338 against 0.359 in relative norm. Mean contact speed runs 0.165 against 0.183 m/s at the emitted state and 0.201 against 0.220 after two steps. Training through the solve lowers the emitted error from 0.367 to 0.314 m and keeps that margin along the solve.

Algorithm 1 Deployed sampling. The solve acts once, on the terminal clean state; the controlled output is the decode of the corrected continuous state, which is never re-quantised.

Require: text c, anchors $\mathcal{A}$, grid $t_0 < \cdots < t_{N-1}$, solver budget R
1: $\boldsymbol{X}_0 \leftarrow$ uniform random tokens
2: **for** $n = 0, \ldots, N-2$ **do**
3: $\quad \pi_n \leftarrow$ anchor-conditioned guided clean-token posterior at $(\boldsymbol{X}_n, c, \mathcal{A})$
4: $\quad \boldsymbol{X}_{n+1} \leftarrow$ first event under rates sampled from π_n ▷ equations 7 and 8
5: **end for**
6: $\pi_{N-1} \leftarrow$ posterior at $(\boldsymbol{X}_{N-1}, c, \mathcal{A})$; $\boldsymbol{X}_N \sim \pi_{N-1}$
7: $\boldsymbol{E}_0 \leftarrow \boldsymbol{C}[\boldsymbol{X}_N]$ ▷ hard clean code embedding
8: $\hat{\boldsymbol{E}} \leftarrow \mathrm{GN}(\boldsymbol{E}_0; \mathcal{A})$, R accepted steps ▷ equation 4, identity metric
9: **return** $\mathcal{D}(\hat{\boldsymbol{E}})$ ▷ decode the corrected state directly

Stage 1 trains the text-conditioned prior: sample $t \sim \mathcal{U}(0,1)$, corrupt the clean token sequence with $q_t(\cdot \mid z^\star)$ from equation 6, and train the transformer to predict $z^\star$ with cross-entropy, dropping the text condition with a fixed probability so that the same network supplies the unconditional branch used by classifier-free guidance. Stage 2 adds the anchor conditioning of §3.6 and fine-tunes the full model, still with cross-entropy alone, over a ladder of anchor densities in which the first and last frames are always observed. Stage 3 is the on-policy calibration. The model generates with its own sampler; anchor conditioning switches on at step 10 of the 20, the last four transitions retain gradients, and the terminal solve runs on the final state inside the graph through equation 9 at $R{=}2$ accepted steps, $\lambda_0{=}10^{-2}$, four trials, a tolerance of 10^{-5} on the mean squared residual and the 48-anchor row cap of the deployed configuration. The objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + 0.1\,\mathcal{L}_{\mathrm{local}}, \qquad \mathcal{L}_{\mathrm{local}} = \overline{\|\mathcal{D}(\boldsymbol{E}_{\mathrm{ST}}) - \boldsymbol{Y}\|}_{\mathcal{A}} + 0.25\,\overline{\|\mathcal{D}(\boldsymbol{E}_{\mathrm{ST}}) - \boldsymbol{Y}\|}_{\mathcal{H}}, \tag{13}$$

with $\mathcal{L}_{\mathrm{CE}}$ the rollout cross-entropy on the sampler's own logits, and $\mathcal{L}_{\mathrm{local}}$ the mean Euclidean joint error of the straight-through decode against the reference clip, averaged over the anchored frame–joint pairs $\mathcal{A}$ and over the halo $\mathcal{H}$ of frames within two of an anchor. The solved state enters the objective through $\mathcal{D}(\boldsymbol{E}_{\mathrm{ST}})$, whose forward value is the corrected state and whose gradient reaches the differentiable posterior state of equation 9. Stage 3 is short: the deployed checkpoint is taken at epoch 25. The path constants of §3.6 are $p{=}2$, $\alpha{=}0.9$ and $c_\beta{=}3.0$, with $N{=}20$ sampler steps on a uniform grid over $[0.1, 0.9]$, the tokens drawn uniformly at random at its first point, so $\dot{\beta}$ is evaluated away from both endpoints of the path, guidance scale 4.0 and top-k truncation to the highest-scoring tenth of the codebook.

Sizes. The masked transformer is 8 layers of width 384 with 6 heads and a 1536-wide feed-forward block, dropout 0.2, and a conditioning drop probability of 0.1 for classifier-free guidance. The frozen tokeniser is a single quantiser over 512 codes of dimension 512; its convolutional decoder is 512 wide and 3 deep with dilation growth 3, and runs at unit temporal stride, neither downsampling nor upsampling in time. The residual encoder reads 54 control features per frame and injects through rank-64 adapters. Clips run to 196 frames. Training uses batch 48 at learning rate $5{\times}10^{-5}$, stepped by 0.1 at 70, 80 and 90 percent of the schedule.