METRIC-LEVEL UNIVERSAL KINEMATIC ATTRIBUTE CONTROL FOR TEXT-TO-MOTION GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-to-motion (T2M) generation has advanced rapidly, yet current systems treat text as a soft semantic specification rather than a metric command: a model can respond to "jump high" but not to "jump 0.4 m", and writing the number into the prompt has virtually no effect. We propose Metric Attribute Guidance (MAG), a training-free, sampling-time method that equips a *frozen* vanilla T2M diffusion model with a metric-level control interface over eight universal, body-intrinsic kinematic attributes, seven of which are enforced by differentiable guidance, commanded in absolute physical units alongside the caption. At each late reverse-diffusion step, MAG differentiably recovers world-space joints from the model's posterior mean, extracts soft attribute proxies, and takes a few band-limited, clipped gradient steps toward the commanded values, requiring no fine-tuning and no backbone modification. Applied *unchanged* to independently trained backbones, MAG raises the mean commanded–realized correlation ρ on HumanML3D from 0.02 to 0.83 under identical commands, with no drop in text fidelity. It transfers to KIT-ML under identical hyper-parameters, reaching centimetre-level median error on the vertical extrema on both benchmarks. These results indicate that metric-level controllability is already latent in standard T2M diffusion models, and can be unlocked purely at sampling time, turning T2M from a semantic sketch into an interface that accepts commands in physical units.

1 INTRODUCTION

Generating realistic and controllable human motion from natural language is a foundational capability for game animation, virtual avatars, embodied AI, and motion analysis. Recent text-to-motion (T2M) systems have made remarkable progress: diffusion-based methods such as MDM (Tevet et al., 2023) and MLD (Chen et al., 2023), discrete-token approaches like T2M-GPT (Zhang et al., 2023a) and MoMask (Guo et al., 2024), and the latest scaled or unified variants (Hwang et al., 2026; Wang et al., 2025b; Guo et al., 2025) can now produce diverse and visually plausible motions from free-form prompts on the HumanML3D (Guo et al., 2022) and KIT-ML (Plappert et al., 2016) benchmarks. In parallel, a rich line of controllable-T2M work adds *spatial* interfaces on top of these backbones, such as root-trajectory keyframes (Karunratanakul et al., 2023; Rempe et al., 2026), arbitrary joint constraints (Xie et al., 2024; Dai et al., 2024), sparse keyframe motion prompts (Zhang et al., 2025; Bae et al., 2025), and even sketched stick figures (Wang et al., 2025a). Meanwhile, fine-grained linguistic methods (Zhang et al., 2023b; Petrovich et al., 2024; Shen et al., 2026) enrich the text interface with body-part-level and timeline-segmented descriptions.

Yet text remains a soft semantic specification rather than a metric command. A user who writes *"jump 0.4 m and reach with the right arm"* obtains a motion that *looks* like jumping and reaching, but nothing ties the realized height to the requested one. Writing the number into the prompt does not help, because training captions almost never carry physical quantities and the model cannot ground them. More data alone would not close this gap. Textual supervision is coarse by construction: captioning pipelines bin continuous values into ordinal tiers ("slow / moderate / fast"), so a numeral is rarely paired with the magnitude it denotes. Where such pairs do occur, the text encoder offers no order-preserving representation of numerical magnitude, and the pairing therefore carries no usable signal. Non-textual interfaces fare no better. Spatial control transfers the specification burden to the user, who must supply a trajectory that realizes 1.5 m/s rather than state the value. Moreover, clip-level quantities such as path length are not localized in any single frame, so sparse keyframes cannot

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.

constrain them, while a dense trajectory over-constrains the motion to fix a single scalar. The closest scalar interface (Dai et al., 2026) specifies ratios between limbs rather than absolute quantities, applies only to quasi-periodic motion, and never verifies that the commanded value is achieved. To our knowledge, no existing work provides a metric-unit, category-agnostic, continuously controllable interface for full-body T2M generation.

We propose Metric Attribute Guidance (MAG) to close this gap. MAG exposes eight universal kinematic attributes with intuitive physical units, each commanded in absolute values alongside the caption. Those attributes are enforced at sampling time by analytic gradient guidance on a *frozen* vanilla T2M diffusion backbone. In summary, this paper makes three main contributions. **(1) A new T2M task formulation: metric-level universal attribute control.** Rather than spatial trajectories or finer-grained language, the user issues a small set of intuitive attribute commands alongside the caption. The attributes are defined once *across* categories, datasets, and skeletons, turning "control" from a data representation problem into a *measurement* problem. **(2) A training-free control method with a narrow backbone interface.** Given a vanilla T2M diffusion checkpoint, MAG enforces commands by gradient guidance at late reverse steps. We apply the same implementation to three independently trained MDM-style backbones covering two skeletons and two feature layouts, so the transfer we demonstrate is across data and representation. **(3) Empirical evidence of the gap and the remedy.** A competent vanilla baseline follows commanded values weakly, and numeric prompting does not help. Under an identical command sweep on the same frozen checkpoint, MAG raises the mean commanded–realized correlation ρ on HumanML3D from 0.02 to 0.83 with no drop in text fidelity, with the same hyper-parameters transferring to KIT-ML and a basketball testbed, where the pattern replicates on the attributes each backbone's training distribution supports.

2 Related Work

2.1 Text-to-Motion Generation

T2M generation is dominated by two model families. *Diffusion-based* methods, including MotionDiffuse (Zhang et al., 2024), MDM (Tevet et al., 2023), and the latent-space MLD (Chen et al., 2023), denoise a motion sequence conditioned on a text embedding (Radford et al., 2021). *Discrete-token* methods, such as T2M-GPT (Zhang et al., 2023a), MoMask (Guo et al., 2024), and ScaleMoGen (Hwang et al., 2026), first quantize motion into tokens and then generate them autoregressively or via masked modeling. Recent work extends this frontier along several axes: scaling to million-clip corpora and motion-language models (Wang et al., 2025b; 2026), improving efficiency (Zeng et al., 2025), rethinking representation and evaluation (Meng et al., 2025), and unifying generation with editing (Guo et al., 2025). We instantiate our method on MDM because its intermediate prediction $\hat{x}_0$ is a complete motion in continuous feature space, which yields a differentiable path to world-space joint positions. Token-based models do not admit such a path, since the quantization and sampling steps that produce discrete indices are non-differentiable. Crucially, all of these backbones treat text as a soft semantic specification, and none exposes a continuous, physical-unit control interface, which is precisely the gap this work addresses.

2.2 Controllable Motion Generation

We group existing control interfaces into three families. They differ substantially in mechanism, and none of them exposes a scalar target in physical units without specific training. *Spatial* methods (Karunratanakul et al., 2023; Xie et al., 2024; Wang et al., 2024; Dai et al., 2024; Zhang et al., 2025; Bae et al., 2025; Wang et al., 2025a) constrain joint or root coordinates at given frames. They can specify where a joint is, but aggregate quantities such as an average speed of 1.5 m/s are integrals over the whole clip that no keyframe can express. *Fine-grained linguistic* methods (Zhang et al., 2023b; Petrovich et al., 2024; Shen et al., 2026) enrich the text with body-part-level or timeline-segmented descriptions, but the control remains qualitative. *Train-time conditioning* fuses the control mapping into the model weights: BPC (Dai et al., 2026) offers per-body-part sliders, but only as unitless relative ratios; ACMo (Wei et al., 2025) and SMooDi (Zhong et al., 2024) require annotated data and a training pass per backbone. In contrast, our interface consists of eight scalars in physical units, and needs no training to use.

2.3 SAMPLING-TIME GUIDANCE

Methodologically, MAG descends from inference-time guidance of diffusion models (Dhariwal & Nichol, 2021), where a differentiable objective steers the intermediate prediction at each reverse step without altering trained weights. This family has since been generalized beyond classifiers to arbitrary losses and energies (Bansal et al., 2024; Yu et al., 2023), and to inverse problems via posterior sampling (Chung et al., 2023). Within T2M, OmniControl (Xie et al., 2024) applies analytic guidance toward spatial joint-position targets, PhysDiff(Yuan et al., 2023) projects intermediate motions toward physical plausibility using a simulator policy, while DNO(Karunratanakul et al., 2024) and ProgMoGen (Liu et al., 2024) optimize user-specified losses or programmed constraints in the latent space. Both are general enough to express a metric attribute target: DNO optimizes differentiable function of the motion, and ProgMoGen provides atomic constraints together with programmatic error functions. What they leave to the user is the objective itself. Our contribution is on that side rather than on the optimizer: a fixed set of unit-carrying attributes portable across datasets and skeletons, a matching commanded-versus-realized protocol, and a placement of the update on the late posterior mean that costs a fraction of noise-space optimization. We make the comparison directly in §4.3 by running DNO with our own attribute loss on the same frozen checkpoint, which isolates the placement of the update from the objective it descends. We supply this missing objective: a small set of differentiable attribute functions reporting values in metric units across motion categories, together with a recovery interface that attaches the same guidance to different backbones and skeletons. Because the quantities are absolute and body-intrinsic, they can be measured back from the output, which yields a commanded-versus-realized fidelity metric and lets one hyper-parameter setting transfer across three backbones, two skeletons, and two feature layouts in this study.

3 METHOD

3.1 PROBLEM SETUP AND OVERVIEW

We represent a motion clip as a joint-position sequence $X = \{p_s^j\}_{s=1..T}^{j=1..J}$ with $p_s^j \in \mathbb{R}^3$, recorded at frame rate ϕ, where $j{=}0$ denotes the root (pelvis). Throughout, s indexes frames and t indexes diffusion steps. A T2M diffusion model G generates X from a caption c. In addition to the caption c, the user supplies a commanded attribute vector $a^* \in \mathbb{R}^M$ together with a mask $m \in \{0,1\}^M$ that selects which attributes to enforce. We then generate $\hat{X} \sim G(c)$ such that the *realized* attributes $f(\hat{X})$ match a^* on the masked entries. Here $f = (f_1, \ldots, f_M)$ is a fixed, analytic, category-agnostic attribute extractor that maps a motion to values in physical units. The method has three defining properties. It is *training-free*: G is a completely frozen vanilla text-only MDM (Tevet et al., 2023) checkpoint, and control is injected purely at sampling time (§3.3). It is *closed-loop*: the f serves as both the control target and the evaluation metric. Finally, it is *skeleton- and dataset-agnostic* and reaches the backbone through a narrow interface: since f operates only on world-space joints, any supported feature layout is mapped to those joints through a differentiable recovery layer $\mathcal{R}$ and a skeleton index table (§4.1, Appendix B). Figure 1 summarizes the resulting sampling loop.

We select M scalar attributes that satisfy four criteria: each must be computable from any $(T, J, 3)$ trajectory, meaningful across action categories, expressed in intuitive physical units, and differentiable or equipped with a smooth differentiable proxy. The resulting set spans four groups. Three are horizontal root aggregates: *speed* (m/s), *path length* (m), and *net displacement* (m). One is an orientation quantity, *facing change*, which we encode as a sin–cos pair to avoid the $\pm 180^\circ$ wrap discontinuity. Three are vertical extrema measured relative to a robust median-height baseline: *jump height* (m), *crouch depth* (m), and *reach height* (m). The last is a periodicity quantity, *tempo* (Hz), defined as the foot-contact gait-cycle frequency. This gives $M = 8$ semantic attributes, counting the sin–cos split, the attribute vector is 9-dimensional. Each attribute also carries an applicability mask, since not every clip exhibits every attribute. Per-dataset z-score statistics are computed over applicable clips only. The details of formal definition and mask criteria are given in Appendix A.

3.2 DIFFERENTIABLE SOFT PROXIES

Gradient guidance requires the attribute map to be differentiable, but several attributes rely on $\max$, $\min$, or median operations that are not smooth. We therefore replace each hard extremum with a

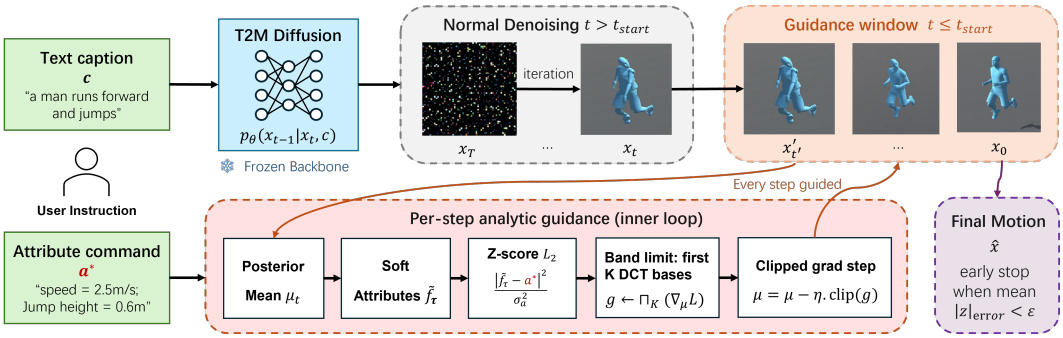


Figure 1: **Method overview.** The caption conditions a frozen vanilla T2M diffusion model, the attribute command is enforced only during late reverse diffusion ($t \leqslant t_{\text{start}}$) by a short inner gradient loop on the posterior mean. The same attribute functions measure the output, closing the specification–measurement loop.

temperature-controlled soft proxy. For the maximum and minimum we use

$$\text{smax}_\tau(x) = \sum_s \text{softmax}(x/\tau)_s \, x_s, \qquad \text{smin}_\tau(x) = -\,\text{smax}_\tau(-x), \tag{1}$$

and for the baseline height we use a soft median, the mean of the middle 30% of sorted values, which is piecewise linear and differentiable except on a measure-zero set of ties. The facing attribute requires no softening, as its sin–cos encoding is smooth by construction. With these substitutions, seven of the eight attributes become differentiable and can be used for guidance. *Tempo* is the exception: it contains a discrete foot-contact count that admits no useful gradient, so we exclude it from guidance and always evaluate it under its hard definition. The temperature τ is annealed with the diffusion step. It starts large and forgiving while the prediction is still noisy, and sharpens in later steps once the extremum should be localized, and the schedule is detailed in Appendix C. Finally, we denote the resulting soft attribute map by $\tilde{f}_\tau$.

3.3 Sampling-Time Analytic Attribute Guidance

Posterior-mean intervention. At reverse step t, the DDPM posterior (Ho et al., 2020) yields a mean μ_t and a variance σ_t^2 in the normalized feature space. Following classifier-style guidance (Dhariwal & Nichol, 2021; Bansal et al., 2024), we modify the posterior mean: within the guidance window, μ_t is updated by a short inner loop of gradient descent on the attribute loss to obtain $\tilde{\mu}_t$, and the reverse step is completed with the original noise variance:

$$x_{t-1} = \tilde{\mu}_t + \sigma_t \, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \tag{2}$$

and outside the guidance window, $\tilde{\mu}_t = \mu_t$ and sampling proceeds unchanged.

Inner loop. We minimize an attribute-matching loss by gradient descent on the posterior mean μ. The loss maps μ back to world-space joints via the motion statistics (μ_X, σ_X) and the differentiable recovery $\mathcal{R}$ (Appendix B), and compares soft attributes with the command in z-score space, where all attributes are commensurate. With attribute statistics (μ_a, σ_a) and the mask m,

$$\mathcal{L}(\mu) = \left\| \left(\frac{\tilde{f}_\tau\big(\mathcal{R}(\mu \cdot \sigma_X + \mu_X)\big) - \mu_a}{\sigma_a} - z^* \right) \odot \bar{m} \right\|_2^2, \qquad z^* = \frac{a^* - \mu_a}{\sigma_a}. \tag{3}$$

Each step band-limits the gradient in time, clips its norm, and descends:

$$g \leftarrow \nabla_\mu \mathcal{L}, \qquad g \leftarrow \Pi_K(g), \qquad g \leftarrow g \cdot \min\left(1, \frac{\gamma_{\max}}{\|g\|_2}\right), \qquad \mu \leftarrow \mu - \eta \, g, \tag{4}$$

where Π_K keeps the first K DCT-II bases of each channel's time series, suppressing high-frequency jitter. The loop stops early once the mean absolute z-score error over active attributes drops below ε. Inner steps follow a late-heavy schedule, favoring the late steps that determine the final clip.

Band-limited gradients. The projection Π_K addresses a failure mode intrinsic to extremum-type attributes. Their soft relaxations (soft max/min over frames) concentrate the gradient almost entirely on the current extremum frame, so repeated inner steps teleport that single frame, e.g. the root at the jump apex, instead of shaping the complete motion. The commanded value is then realized, but by a physically invalid single-frame spike, which a per-clip realized measurement cannot distinguish from a genuine jump. Projecting g onto the first K DCT bases caps the temporal frequency the guide can inject at roughly $K\phi/2T$; with $K{=}12$ over a 60-frame clip at $\phi{=}20$ fps, this is about 2 Hz. No frame can then be displaced in isolation, sine each correction is spread over its temporal neighborhood, and different actions move together. Low-frequency DCT bases are moreover a classical compact representation of human motion trajectories (Mao et al., 2019). The projection is attribute-agnostic, a single linear operator applied to whatever gradient the loss produces, and adds only two $K{\times}T$ contractions per inner step, against a basis precomputed once. Empirically, it removes the spikes on the sustained-pose extremum attributes and improves their control (§4.5).

Guidance schedule. MAG is governed by four hyper-parameters: the inner step size η, the delayed start t_{start}, the gradient clamp $\gamma_{\max}$, and the band limit K. For $t > t_{\text{start}}$, the frozen backbone denoises *unguided*, establishing the caption-conditioned motion structure. Guidance then adjusts the numeric targets during the second half of sampling, without fighting the text condition. Removing the delay alone ($t_{\text{start}}{=}999$) already costs the worst-attribute correlation and producing visible root drift on inspection. Reversing all three schedule parameters together collapses control outright, as our ablation shows (§4.5). Classifier-free guidance (Ho & Salimans, 2021) on text is kept as in the base model. Guidance requires inner-loop backward passes only through $\tilde{f}_\tau \circ \mathcal{R}$, a closed-form kinematic map rather than the diffusion network itself, which keeps the added sampling cost modest (Table 2), and it is paid only when metric control is requested. The identical configuration is used on every dataset and backbone in this paper, with no per-dataset tuning.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate MAG on two public benchmarks. HumanML3D (Guo et al., 2022) serves as the primary benchmark and comprises 14,616 motions with 44,970 captions at 20 fps. Mirroring augmentation yields the standard split of 23,384 / 1,460 / 4,384 clips for training, validation, and testing. KIT-ML (Plappert et al., 2016) provides a second benchmark (3,911 motions, 6,278 captions, resampled to 20 fps) with a different skeleton and feature layout, and thus allows us to verify that a single guidance configuration transfers to an independently trained backbone. We additionally maintain an internal testbed, Basketball, whose per-frame ground-truth (GT) permits direct validation of the attribute extractor. As this dataset is not publicly available, the corresponding results are confined to Appendix G, and all conclusions in the main paper rest on the two public benchmarks.

Metrics. We report two families of metrics. For *quality*, we use R-Precision and MM-Dist (Guo et al., 2022), computed by a per-dataset two-tower evaluator trained from scratch with an InfoNCE objective (Oord et al., 2018). Testing models are compared under the same evaluator, so the comparison is internally consistent. For *physical fidelity*, the attribute extractor f can serve as the measurement instrument. For a commanded value c_k and a generated clip $\hat{X}$, the realized value is $r_k = f_k(\hat{X})$. Commanded targets are drawn from the attribute's percentile statistics over the training split rather than copied from the reference clip, so the protocol measures open-loop command following rather than reconstruction. Fixing a caption, we sweep c_k over 9 levels and generate $N{=}8$ samples per level, then report three quantities per attribute: the *Pearson correlation* $\rho(c_k, r_k)$, which tests whether realized values track commands at all. The *median absolute error* P_{50}, which gives the typical deviation in physical units. And the *control-curve slope* α from a fit $r = \alpha c + \beta$, which measures response rate and equals 1 for an ideal controller. A command is physically meaningful if the data manifold contains motions that realize it, so the levels are placed by percentile of the attribute's distribution. Five of the nine lie within $[p_{20}, p_{80}]$, two probe below the twentieth percentile, and the outermost two, which we call *stress levels*, sit in the sparse upper tail, the last of them at the largest value the split contains (Appendix F lists the placement details).

Compared models. We ablate two design choices, train-time attribute conditioning and sampling-time guidance, yielding four variants of the same architecture: **vanilla MDM**, the text-only back-

bone; **Attribute Condition (AttrCond)**, which learns an attribute-conditioning pathway at training time; **AttrCond+G**, which adds guidance on top of AttrCond; and **MAG**, our main method, which applies guidance to the vanilla backbone and thus injects attributes only at sampling time. Comparing MAG with AttrCond isolates sampling-time from train-time injection, and comparing it with AttrCond+G tests whether the two mechanisms compound. We further include two external baselines: **OmniControl** (Xie et al., 2024), the representative spatial-control method, evaluated on every attribute expressible through its *(joint, frame, xyz)* interface; and **DNO** (Karunratanakul et al., 2024), the representative noise-optimization framework, run with our attribute loss (Eq. 3) on the same frozen checkpoint, so that the two methods differ only in the optimization mechanism.

Implementation. All backbones are vanilla MDM (Tevet et al., 2023) transformer encoders trained separately per dataset, architecture and training hyper-parameters are listed in Appendix C. A single guidance implementation is shared across all backbones: each feature layout is adapted through the differentiable recovery $\mathcal{R}$, a skeleton index table, and a unit-scale factor (Appendix B). The same configuration is used for every dataset and every variant without per-dataset tuning.

4.2 Two Routes to Metric Control, and Why Both Fall Short

Text control. The vanilla MDM reaches R@1 $= 0.450$, about 80% of the evaluator's ceiling (Table 1). Its attribute behaviour is far weaker, and two distinct tests make this precise. First, without any command, we generate one clip per test caption and compare its attributes against those of the corresponding reference clip: the correlation averages $\rho = 0.35$, so the caption carries an ordinal signal that narrows the plausible range without fixing the value. Second, and this is the comparison we carry through the paper, we write the value into the caption (*"A person jumps exactly 0.35 meters high."*), vary only the number, and score the result with the control-curve protocol used for methods in Table 2. Here the mean correlation is $\rho = 0.02$ and the α is indistinguishable from zero on seven of eight attributes. Therefore, the second is a command-following baseline, since the first asks whether an unprompted sample lands near its reference while the second asks whether a number in the caption is obeyed. Each guided or conditioned row in Table 1 is generated by commanding every clip its own GT attribute vector, restricted to the attributes that clip exhibits, so the text metric is measured under commands drawn from the data distribution rather than from the sweep of Table 2.

Table 1: R-Precision on HumanML3D test. OmniControl uses a different backbone and feature space, so its R-Precision is not comparable under our protocol.

Model	R@1↑	R@3↑	MM-Dist↓
Random	0.031	0.094	—
Vanilla MDM	0.450	0.713	1.018
AttrCond	0.438	0.690	1.022
AttrCond (text-only)	0.202	0.364	1.255
AttrCond+G	0.443	0.699	1.021
MAG (Ours)	**0.469**	**0.727**	**1.004**
DNO	0.408	0.640	1.048
Ground Truth	0.561	0.787	0.933

Train-time conditioning. AttrCond concatenates the commanded attribute vector and its applicability mask to the text embedding and continues training from the vanilla checkpoint. Supplying the attributes does improve controllability, raising mean-ρ on the command sweep from 0.02 to 0.64, but two drawbacks remain. First, the model **becomes reliant on the attribute input**: in text-only sampling, mean-ρ falls to 0.15 and R@1 drops to 0.202 (Table 1), since the attribute pathway is present at training step and removing it at inference places the model outside its training distribution. Second, its control is **learned rather than enforced**. AttrCond overshoots speed while barely responding to path length and net displacement (Table 2), a pattern that follows what captions can supply: speed is strongly coupled to the verb, whereas clip-level aggregates such as path length have no lexical correlate and therefore receive a weak condition signal during training. Dai et al. (2026) also reported a comparable neglect of parts of the condition input. **MAG takes a third route**, it leaves the text-only backbone untouched, so text-only behaviour is unaffected, and descends each command at sampling time rather than inferring it from correlations in the train data.

4.3 Main Results on HumanML3D

Text fidelity is preserved. As shown in Table 1, MAG reaches R@1 $= 0.469$ on R-Precision, slightly above the vanilla model's 0.450, with matching MM-Dist: sampling-time guidance does not hurt caption–motion alignment. In that table, each clip is commanded with attributes measured from

Table 2: **Control accuracy on HumanML3D.** Each cell reports α / ρ / P_{50} over a sweep of 9 commanded levels with 8 samples each, and P_{50} is in the attribute's natural unit (m, deg, or Hz). Facing change is periodic, so we compute ρ on $[\sin c, \cos c]$ against $[\sin r, \cos r]$ to avoid the $\pm 180^\circ$ wrap, and P_{50} as a circular difference in degrees. The former does not bound the latter, so both are needed. OmniControl entries marked "—" are attributes inexpressible as its constraints; * marks its *one-sided* response. Mean-ρ is computed over each method's expressible attributes, which are not the same set in every column: OmniControl has five, the remaining methods seven, so the means are not comparable across columns and we report min-ρ in the text for that purpose. Jitter (mean joint acceleration) and Peak Jerk are averaged over the in-range reach sweep.

Attribute	OmniControl	DNO	AttrCond	AttrCond+G	**MAG (Ours)**
Speed	0.79 / 0.99 / 0.04	0.84 / 1.00 / 0.20	1.51 / 0.92 / 0.20	1.41 / 0.96 / 0.11	0.79 / 0.92 / 0.49
Path length	—	0.31 / 0.98 / 1.05	0.09 / 0.69 / 0.94	0.47 / 0.99 / 0.89	0.42 / 0.97 / 1.34
Net displace	1.00 / 1.00 / 0.00	0.73 / 1.00 / 0.31	0.04 / 0.35 / 0.97	0.21 / 0.86 / 0.77	0.52 / 0.97 / 0.60
Facing change	—	0.74 / 0.71 / 33.4	0.06 / 0.07 / 84.7	0.10 / 0.18 / 75.9	0.40 / 0.63 / 45.6
Jump height	0.63 / 0.93 / 0.22*	0.43 / 0.95 / 0.039	0.46 / 0.84 / 0.103	0.80 / 0.95 / 0.021	0.58 / 0.76 / 0.037
Crouch depth	0.63 / 0.71 / 0.084*	0.24 / 0.82 / 0.093	0.36 / 0.84 / 0.103	0.81 / 0.93 / 0.048	0.49 / 0.54 / 0.102
Reach height	0.60 / 0.90 / 0.204*	0.72 / 0.98 / 0.017	0.48 / 0.80 / 0.217	0.97 / 0.99 / 0.028	0.89 / 0.98 / 0.018
Tempo	—	—	−0.20 / −0.39 / 0.69	−0.20 / −0.39 / 0.69	0.06 / 0.10 / 1.21
$\rho \geq 0.5$	5/8	**7/8**	5/8	6/8	**7/8**
Mean-ρ	0.910 (5 Attr.)	**0.920** (7 Attr.)	0.644 (7 Attr.)	0.837 (7 Attr.)	0.826 (7 Attr.)
Sec/Clip	14.7	63.9 (100-step)	**1.8**	9.6	8.8 / 2.8 (100-step)
Jitter (m/s^2)	**0.38**	3.45	0.69	0.67	0.91
Peak Jerk (m/s^3)	7072	1147	**106**	109	221

its own reference motion, so retrieval could in principle be helped by information leaking from that reference. We therefore repeat the measurement with commands that carry no such information, and the results shows that R@1 does not drop (details in Appendix F). Both AttrCond variants fall below the backbone they were fine-tuned from, a consequence of the trained attribute pathway rather than of guidance tuning. DNO loses text fidelity, even though it pursues the same attribute objective on the same frozen backbone. Hundreds of gradient steps in noise space drift the sample off the text-conditioned manifold.

Control accuracy. Table 2 reports control accuracy for all methods and Figure 2 shows the corresponding curves. To avoid comparing against an under-tuned baseline, AttrCond+G is evaluated at MAG's own guidance configuration. At the best setting we found for it (Appendix C), AttrCond+G attains a slightly higher mean-ρ than MAG. It is also has more accurate P_{50} on four attributes, including jump (2.1 against 3.7 cm) and crouch (4.8 against 10.2 cm). Among the four variants, MAG is the only one with $\rho \geq 0.5$ on every differentiable attribute, whereas AttrCond+G collapses on facing change. Since the reliability of a control interface is largely set by its weakest attribute, we report min-ρ alongside the mean throughout. Train-time conditioning plus guidance is at least as accurate as guidance alone and smoother, provided a retraining pass is affordable and the attribute set is fixed in advance. Guidance alone reaches comparable mean accuracy with a better worst case, and requires neither. Every accuracy column is at DDPM-1000 except DNO, which we run with a 100-step DDIM that is a default configuration of the method (Appendix I).

The OmniControl results reveal how control degrades as attributes move away from its native interface. As expected, attributes that coincide with frame–joint coordinates (net displacement, speed) are near-perfect under geometric constraints, because OmniControl can pin those coordinates directly. Clip-level extrema (jump, crouch, and reach heights) instead show a *one-sided* response: pinning a single frame can raise the extremum but cannot cap it, since unconstrained frames remain free to overshoot the target (details in Appendix H). Finally, aggregates, orientation, and periodicity cannot be expressed as (joint, frame, xyz) constraints at all.

The DNO comparison isolates the *optimization mechanism*: DNO and MAG run the same checkpoint, loss, and early stop, and differ only in what they optimize. DNO optimizes the initial noise, so its gradient passes through the full 100-step DDIM unroll. MAG optimizes the posterior mean, so its gradient passes only through the closed-form kinematic map. On accuracy the two mechanisms are comparable, but they diverge sharply on cost and physical quality. Backpropagating through the 100-step unroll at every descent step costs DNO an average of 63.9 s per clip, against MAG's 2.8 s on a matched 100-step sampler. More fundamentally, DNO's clips carry transient single-frame chain collapses of up to -29% at in-range reach commands (Appendix I, Figure 6). These are invisible to

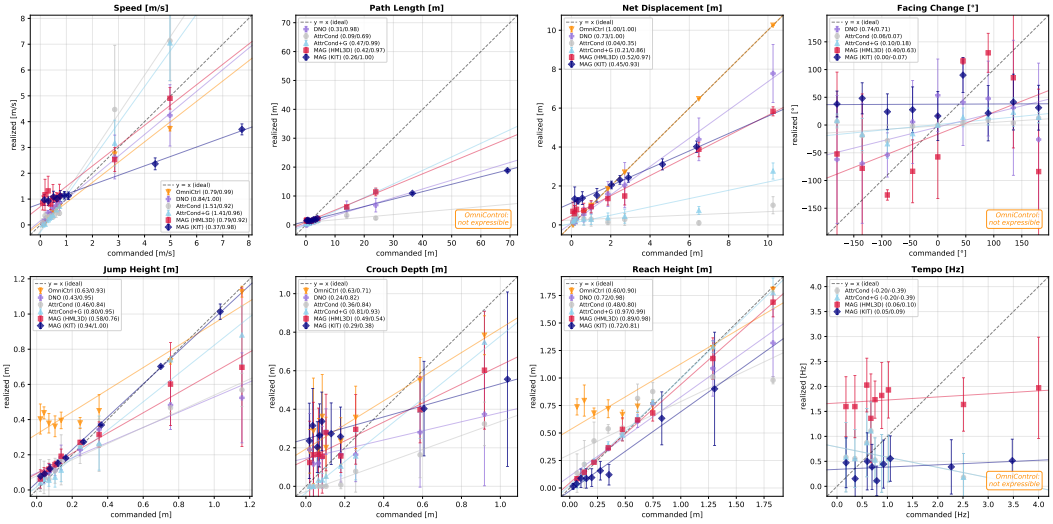


Figure 2: **Control curves.** Each panel plots the realized value of an attribute against the commanded value, with the α and ρ of the fitted relation given in the legend.

whole-clip accuracy, which reads off a single extremum, but show up in Jitter and Peak jerk. The same optimization difficulty limits how far DNO converges: on the full-attribute target it averages 147 descent steps and still hits the 300-step cap on 37% of batches without reaching the $\epsilon = 0.05$.

Feasible range vs. Stress levels. The stress levels sit in the sparse upper tail of the training distribution. Guidance can only meet such commands by leaving the main data manifold, and it does: the arm chain stretches by up to $+38\%$ there, against a maximum of $+3.6\%$ at the interior levels (Appendix J). We keep them in every aggregate rather than reporting a flattering subset, and treat the interior of the distribution as the operating specification of MAG. Resolving the error by level shows that the two regimes are separated by a break rather than a gradient: median error is flat across $[p_{20}, p_{80}]$ and stays at or below 0.043 m for jump height and 0.044 m for reach height throughout, then rises abruptly at the largest command, where jump error goes from 0.030 to 0.413 m, more details are given in Appendix F and Table 13. A percentile is in any case a screen rather than a certificate, since it is computed by pooling all captions.

Qualitative Results. Figure 3 makes the central claim visible on a frozen vanilla checkpoint. When the value is spelled out in the prompt, the realized jump height is flat and uncorrelated with the command. When the caption is held fixed and the same value is routed through guidance as a scalar command, the realized height rises monotonically. A reach-height sweep on the same checkpoint tracks the command even more closely, which is shown in Appendix F. Additionally, Appendix G reports qualitative sweeps on the basketball testbed.

4.4 Generality Across Backbones

To test whether MAG transfers across backbones, we train a vanilla MDM on KIT-ML dataset from scratch and apply MAG with unchanged hyper-parameters. The pattern observed on HumanML3D replicates, as shown in Table 3 and Figure 2. Text fidelity is preserved, with R@1 $= 0.513$ against the vanilla model's 0.511. Jump-height control is near-perfect, with a median error of 2.3 cm. Five of the seven differentiable attributes remain controllable ($\rho \geq 0.5$) on KIT-ML, and the mean-ρ across those seven is 0.72. Crouch depth and facing change are weaker, which we attribute to data support: KIT-ML contains few such motions for guidance to exploit, consistent with the principle that guidance steers

Table 3: MAG control accuracy on KIT-ML, with unchanged hyper-parameters.

Attribute	α	ρ	P_{50}
Speed	0.37	0.98	0.53
Path length	0.26	1.00	1.09
Net disp.	0.45	0.93	0.71
Facing change	0.00	-0.07	95.8
Jump height	0.94	1.00	0.023
Crouch depth	0.29	0.38	0.202
Reach height	0.72	0.81	0.047
Tempo	0.05	0.09	0.70

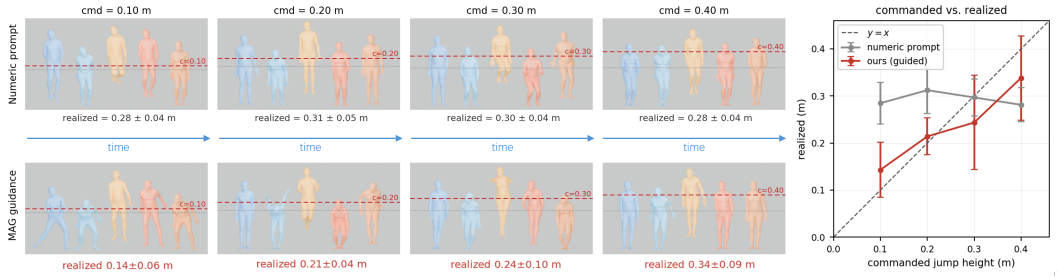


Figure 3: **Jump-height sweep on one frozen vanilla MDM** (caption *"A person performs a jump."*, commanded 0.10–0.40 m). Top row: the value is written into the prompt (*"jumps exactly v meters high"*) and the realized height stays flat. Bottom row: the caption is fixed and the value is supplied as a scalar command through our method, and the realized height rises with the command. Each cell shows keyframes around the jump apex of the median-realized sample. The right panel plots commanded against realized height, and the annotations report mean±std over 8 samples.

within the backbone's learned manifold. An in-the-wild validation on a Basketball testbed, including GT calibration of the extractor and guidance on an independently trained backbone, is reported in Appendix G.

4.5 Ablation

Table 4 ablates the guidance parameters. The band limit K sits on a wide plateau: $K \in \{12, 24\}$ are within 0.003 mean-ρ of each other, so we fix $K{=}12$ as a trade-off choice, without per-dataset tuning. $K{=}6$ degrades min-ρ and extremum-attribute precision, as the band is too narrow for the guidance to shape a take-off. $K{=}0$ brings back the single-frame spikes discussed in §3.3. The three schedule parameters (η, t_{start}, $\gamma_{\max}$) function as a joint system: each single reversal harms a different axis, yet mean-ρ stays within 0.01 of the reference in all three cases, because easy attributes such as reach and jump mask failures in the worst one. Therefore, we use min-ρ and P_{50} as the operative selection criteria. Reversing all three at once collapses control, with min-ρ falling to 0.135. This collapse is invisible to standard metrics. R-Precision is flat across every guidance configuration, and FID is not merely flat but ordered inversely to controllability. Only the commanded–realized protocol registers the failure (Appendix J).

Table 4: Ablation on HumanML3D. Upper block: band limit K at the main schedule ($\eta{=}0.1$, $t_{\text{start}}{=}500$, $\gamma_{\max}{=}2$). Lower block: one schedule parameter reversed at a time, run without projection and therefore referenced against the no-projection row.

Variant	R@1	mean-ρ	min-ρ	speed P_{50}
$K{=}12$ (default)	0.469	0.826	0.548	0.49
$K{=}6$	0.465	0.822	0.482	0.59
$K{=}24$	0.460	0.823	0.523	0.45
$K{=}0$	0.453	0.810	0.499	0.33
A: $\eta{=}0.3$	0.458	0.810	0.504	0.93
B: $t_{\text{start}}{=}999$	0.462	0.809	0.419	0.32
C: $\gamma_{\max}{=}5$	0.455	0.808	0.474	0.70
A+B+C	0.464	0.587	0.135	3.02

5 Conclusion

We presented Metric Attribute Guidance, a training-free analytic guidance method that equips a frozen text-to-motion diffusion model with a metric-level interface over eight universal kinematic attributes, seven of which admit differentiable guidance, with no attribute-conditioning training and no architectural change. Applied unchanged to independently trained backbones spanning two skeletons and two feature layouts, MAG raises the mean control correlation on HumanML3D from $\rho = 0.02$ to 0.83 under an identical command sweep, with no drop in text fidelity, and attains centimeter-level accuracy on the vertical extrema of KIT-ML under identical guidance hyperparameters. Guidance steers within the distribution a backbone has already learned, which bounds what any command can achieve. We therefore see it as a property worth measuring in text-to-motion systems rather than a capability that must be trained in.

AI USE STATEMENT

We used generative AI tools in three parts of this work. In manuscript preparation, AI tools assisted with polishing the text. In data processing, they assisted with writing scripts for corpus organization, attribute extraction, and result aggregation. In implementation, they assisted with parts of the code base, primarily boilerplate and utility routines. All AI-assisted code and scripts were reviewed and tested by the authors, and every number reported in the paper was produced by code we verified. AI tools did not design the method, choose the experiments, or interpret the results, the research questions, technical content, claims, and conclusions are the authors' own. We have reviewed all AI-assisted work and take full responsibility for the final content of this paper, including text, code, and released artifacts.

REFERENCES

Jinseok Bae, Inwoo Hwang, Young-Yoon Lee, Ziyu Guo, Joseph Liu, Yizhak Ben-Shabat, Young Min Kim, and Mubbasir Kapadia. Less is more: Improving motion diffusion models with sparse keyframes. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11069–11078. IEEE, 2025.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Roni Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 51304–51323, 2024.

Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18000–18010. IEEE, 2023.

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*, 2023.

Minyue Dai, Ke Fan, Anyi Rao, Jingbo Wang, and Bo Dai. Controllable text-to-motion generation via modular body-part phase control. *arXiv preprint arXiv:2603.19795*, 2026.

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 *European Conference on Computer Vision*, pp. 390–408. Springer, 2024.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 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. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5142–5151. IEEE, 2022.

Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1900–1910. IEEE, 2024.

Ziyan Guo, Zeyu Hu, De Wen Soh, and Na Zhao. Motionlab: Unified human motion generation and editing via the motion-condition-motion paradigm. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13869–13879, 2025.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Inwoo Hwang, Hojun Jang, Bing Zhou, Jian Wang, Young Min Kim, and Chuan Guo. Scalemogen: Autoregressive next-scale prediction for human motion generation, 2026.

Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. Guided motion diffusion for controllable human motion synthesis. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2151–2162. IEEE, 2023.

Korrawe Karunratanakul, Konpat Preechakul, Emre Aksan, Thabo Beeler, Supasorn Suwajanakorn, and Siyu Tang. Optimizing diffusion noise can serve as universal motion priors. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1334–1345. IEEE, 2024.

Hyunjun Kim, Gwanghyun Kim, and Se Young Chun. Lamogen: Laban movement-guided diffusion for text-to-motion generation. *arXiv preprint arXiv:2509.24469*, 2025.

Hanchao Liu, Xiaohang Zhan, Shaoli Huang, Tai-Jiang Mu, and Ying Shan. Programmable motion generation for open-set motion control tasks. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1399–1408. IEEE, 2024.

Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9488–9496. IEEE, 2019.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27859–27871. IEEE, 2025.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Mathis Petrovich, Or Litany, Umar Iqbal, Michael J Black, Gul Varol, Xue Bin Peng, and Davis Rempe. Multi-track timeline control for text-driven 3d human motion generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1911–1921, 2024.

Matthias Plappert, Christian Mandery, and Tamim Asfour. The kit motion-language dataset. *Big Data*, 4(4):236–252, 2016.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

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

Keming Shen, Bizhu Wu, Junliang Chen, Xiaoqin Wang, and Linlin Shen. Finextrol: Controllable motion generation via fine-grained text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 8905–8913, 2026.

Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations*, 2023.

Tao Wang, Zhihua Wu, Qiaozhi He, Jiaming Chu, Ling Qian, Yu Cheng, Junliang Xing, Jian Zhao, and Lei Jin. Stickmotion: Generating 3d human motions by drawing a stickman. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12370–12379. IEEE, 2025a.

Ye Wang, Sipeng Zheng, Bin Cao, Qianshan Wei, Weishuai Zeng, Qin Jin, and Zongqing Lu. Scaling large motion models with million-level human motions. In *Forty-second International Conference on Machine Learning*, 2025b.

Yuan Wang, Di Huang, Yaqi Zhang, Wanli Ouyang, Jile Jiao, Xuetao Feng, Dan Xu, and Shixiang Tang. Motiongpt-2: A general-purpose motion-language model for motion generation and understanding. *IEEE Transactions on Circuits and Systems for Video Technology*, 2026.

Zhenzhi Wang, Jingbo Wang, Yixuan Li, Dahua Lin, and Bo Dai. Intercontrol: Zero-shot human interaction generation by controlling every joint. In *NeurIPS*, 2024.

Mingjie Wei, Xuemei Xie, and Guangming Shi. Acmo: Attribute controllable motion generation. *arXiv preprint arXiv:2503.11038*, 2025.

Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. Omnicontrol: Control any joint at any time for human motion generation. In *International Conference on Learning Representations*, volume 2024, pp. 28176–28194, 2024.

Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23117–23127. IEEE, 2023.

Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. Physdiff: Physics-guided human motion diffusion model. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15964–15975. IEEE, 2023.

Ling-An Zeng, Guohong Huang, Gaojie Wu, and Wei-Shi Zheng. Light-t2m: A lightweight and fast model for text-to-motion generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9797–9805, 2025.

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. T2m-gpt: Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023a.

Mingyuan Zhang, Huirong Li, Zhongang Cai, Jiawei Ren, Lei Yang, and Ziwei Liu. Finemogen: Fine-grained spatio-temporal motion generation and editing. *Advances in Neural Information Processing Systems*, 36:13981–13992, 2023b.

Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *IEEE transactions on pattern analysis and machine intelligence*, 46(6):4115–4128, 2024.

Zongye Zhang, Bohan Kong, Qingjie Liu, and Yunhong Wang. Towards robust and controllable text-to-motion via masked autoregressive diffusion. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 9326–9335, 2025.

Lei Zhong, Yiming Xie, Varun Jampani, Deqing Sun, and Huaizu Jiang. Smoodi: Stylized motion diffusion model. In *European Conference on Computer Vision*, pp. 405–421. Springer, 2024.

A Formal Definitions of the Eight Kinematic Attributes

This part gives the formal definitions summarized in §3.1. Throughout, $r_s = p_s^0$ is the root position, superscripts x, y, z pick coordinates (y up), and r_s^{xz} is the horizontal root projection. The vertical baseline is the *median* root height $b(X) = \mathrm{median}_s\, r_s^y$, which is more robust than the first frame against clips that begin mid-air or mid-crouch. It carries an assumption worth stating: that the clip spends most of its frames near a neutral stance, so that the median approximates it. Where that fails the measurement follows the definition rather than the intent. A clip held in a deep crouch for most of its frames and rising briefly at the end takes the crouched height as its baseline, and the rise is then recorded as jump height while crouch depth reads zero, even with the feet never leaving the ground. Trimming the same motion differently shifts the median and with it both values. The attributes are therefore well defined as descriptors of root displacement relative to a clip's own typical height, which is what the guidance descends and what the evaluation measures, and they coincide with the everyday sense of jumping and crouching only for clips whose neutral stance dominates. The extractor calibration of Appendix G is run on clips of this kind.

Horizontal root aggregates.

$$f_{\text{speed}} = \frac{\phi}{T-1} \sum_{s=1}^{T-1} \left\| r_{s+1}^{xz} - r_s^{xz} \right\|_2 \tag{5}$$

$$f_{\text{path}} = \sum_{s=1}^{T-1} \left\| r_{s+1}^{xz} - r_s^{xz} \right\|_2, \qquad f_{\text{disp}} = \left\| r_T^{xz} - r_1^{xz} \right\|_2 \tag{6}$$

Facing change. The heading ψ_s is derived from the hip axis: $v_s = p_s^{\text{RHip}} - p_s^{\text{LHip}}$, $\psi_s = \mathrm{atan2}(-v_s^z,\, v_s^x) - 90^\circ$ (the true forward direction is the right-hip axis rotated by -90°). Because heading is periodic, we encode the first-to-last circular difference as a sin–cos pair, avoiding the $\pm 180^\circ$ wrap discontinuity in both the loss gradient and the dataset statistics:

$$f_{\text{face}} = \big(\sin \Delta\psi,\ \cos \Delta\psi\big), \qquad \Delta\psi = \mathrm{wrap}(\psi_T - \psi_1) \in (-180^\circ, 180^\circ] \tag{7}$$

Vertical attributes. With baseline $b(X)$:

$$f_{\text{jump}} = \max\big(0,\ \max_s r_s^y - b(X)\big), \quad f_{\text{crouch}} = \max\big(0,\ b(X) - \min_s r_s^y\big) \tag{8}$$

$$f_{\text{reach}} = \max_{s,\, j \in \{\text{LHand,RHand}\}} p_s^{j,y} \ - \ b(X) \tag{9}$$

Tempo. We detect foot contacts per frame at the four foot joints (ankles and toes). A joint is in contact when

$$\mathrm{contact}_s^j = \mathbb{1}\big[\, h_s^j < 0.08\ \mathrm{m} \ \wedge\ \|\dot{p}_s^{j,xz}\| < 0.5\ \mathrm{m/s}\,\big], \tag{10}$$

where $\dot{p}_s^{j,xz}$ is the velocity in the ground plane computed by finite differences scaled by the frame rate. A side is in contact if either of its two joints is, $c_s^{\text{side}} = \mathrm{contact}_s^{\text{ankle}} \vee \mathrm{contact}_s^{\text{toe}}$. Counting the $0{\to}1$ rising edges of c_s^{side} gives the number of gait cycles on that side, $n_{\text{cycles}}^{\text{side}}$, and

$$f_{\text{tempo}} = \frac{\frac{1}{2}\big(n_{\text{cycles}}^L + n_{\text{cycles}}^R\big)}{T/\phi}, \tag{11}$$

with T is the clip length in frames and ϕ is the frame rate.

Applicability masks and statistics. The eight attributes occupy nine dimensions, since facing change is encoded as a sin–cos pair. Each attribute carries an *applicability mask* recording whether a clip exhibits it at all: a clip whose hip never rises 5 cm above baseline has no jump height to speak of, so $m_{\text{jump}} = 0$. Thresholds are 0.05 m for jump, crouch and reach, 0.2 m/s for speed, and 0.5 Hz for tempo. The remaining attributes are defined for every clip. The mask is used in three places. Per-dataset statistics (μ_k, σ_k) are accumulated over applicable clips only, since including inapplicable ones as zeros would bias μ_k downwards and inflate σ_k. During sampling, the guidance objective is restricted to the attributes the user commands, which we denote $\mathcal{A}$ and treat as a separate quantity from the data-side mask. In evaluation, an attribute is scored whenever it is either exhibited or commanded, $m_k^{\text{eval}} = \max\big(m_k,\ \mathbb{1}[k \in \mathcal{A}]\big)$, so a command that the model fails to realize is recorded as a large error rather than discarded as inapplicable. The sin–cos pair already lies in $[-1, 1]$ and is not z-scored.

B Model- and Skeleton-Agnostic Instantiation

The guidance interacts with a backbone mainly through two objects: a differentiable map $\mathcal{R}$ from the backbone's feature space to world-space joint positions $(T, J, 3)$, and a skeleton index table naming the joints that the attribute extractor reads. We instantiate both for two feature layouts that share neither dimensionality nor parameterisation, and for two skeleton topologies of different size. The two layouts also differ in skeleton. Basketball dataset was captured on an in-house rig, from which we take the 22 body joints. We bring HumanML3D onto the same joint set and ordering by recovering per-joint rotations from the released 22-joint positions with inverse kinematics, followed by an explicit index remap. And the same feature layout and joint set therefore serve both corpora. KIT-ML provides its own 21-joint topology, which we use without modification. The guidance itself is unaffected by either decision, since all topology-dependent information is confined to the index table described below.

Joint-position layout. This layout has 22 joints with 273-dimension, which is used for HumanML3D and Basketball dataset. Features store the root world position together with joint positions expressed relative to the root, so recovery reduces to a slice-and-add, $\mathcal{R}(x) = jp + rp$, with rp broadcast over the joint axis. The layout is $rp(3) + ra(2) + jp(66) + jv(66) + ja(132) + f(4)$: root position, root rotation, joint positions, joint velocities, 6D joint rotations, and binary foot-contact flags.

HumanML3D-style layout. This layout has 21 joints with 251-dimension, which is instantiated for KIT-ML dataset. Features store root *velocities* rather than absolute root state: angular velocity $\dot{\theta}_s$, planar linear velocity v_s^{xz} in the root-aligned frame, and root height r_s^y. The dimensionality follows from the joint count J as $4{+}3(J{-}1){+}6(J{-}1){+}3J{+}4$, which gives 251 for KIT-ML's 21 joints; the same expression yields the 263-dim, 22-joint variant carried by public HumanML3D checkpoints, and because the recovery below is written in terms of J, that variant needs no additional code. The KIT-ML layout is $r_{\text{vel}}(1) + l_{\text{vel}}(2) + r_y(1) + ric(60) + rot6d(120) + v(63) + f(4)$, where the ric block holds the $J-1$ non-root joints. Recovery integrates the root trajectory and maps the joint offsets back to the world frame:

$$\theta_s = \sum_{u<s} \dot{\theta}_u, \qquad r_s^{xz} = \sum_{u<s} R(\theta_u)\, v_u^{xz}, \qquad p_s^0 = \big[r_s^x,\, r_s^y,\, r_s^z\big], \qquad p_s^j = R(\theta_s)\, ric_s^j + \big[r_s^x,\, 0,\, r_s^z\big] \tag{12}$$

for $j = 1, \ldots, J-1$, with $\theta_0 = 0$ and $r_0^{xz} = 0$. Here $R(\theta)$ is the yaw rotation taking the root-aligned frame to the world frame, and the velocity channel at frame u describes the displacement from u to $u{+}1$. Note that the ric offsets are root-relative in the ground plane but retain absolute height, so the root height enters only through p_s^0 and is read directly from r_y rather than integrated. Every operation involved is smooth (cumulative sums and quaternion rotations), so gradients propagate from the attribute losses to the backbone features end to end.

Two details govern portability. First, the skeleton table carries the five index groups required by the extractor, namely the left and right hip joints (whose axis defines the facing direction), the left and right hand joints, and the group of four foot joints (both ankles and both toes). The 22-joint and 21-joint topologies consequently share a single process path. Second, a scalar unit scale reconciles feature units with attribute statistics, since KIT-ML features are stored in millimetres while attributes are defined in metres. Beyond these two conventions the guidance makes no assumption about absolute placement. In addition to a differentiable feature-to-joints map and a skeleton index table, porting MAG to a new backbone requires the unit scale, which is always known.

C Hyper-parameters

Throughout, t denotes the diffusion timestep, counting down from $T = 1000$ to 0, so smaller t means a cleaner sample. MAG is defined by four parameters, kept identical across all datasets and backbones in this study. The per-sample early-stop tolerance is $\varepsilon = 0.05$, measured as the mean absolute z-score error over the active attributes. The inner loop uses a late-heavy schedule that concentrates its budget where the final clip is determined: 10 steps for $t \geq 100$, 20 for $t < 100$, 100 for $t < 20$, and 500 for $t < 10$. The largest budget is inexpensive in practice, since the early stop terminates most runs well before it is reached. The soft-proxy temperature is annealed along with t: $\tau = 0.1$ for $t \geq 100$, 0.05 for $20 \leq t < 100$, and 0.02 for $t < 20$. The proxy is thus kept coarse

and forgiving while the prediction is still noisy, and sharpened once the extremum can be localized. Classifier-free guidance on the text uses scale 2.5, as in the base model. AttrCond+G is run with this same configuration on the attribute-conditioned backbone. We verified it to be the best setting among those we tried, so the baseline is compared at its own strongest operating point rather than at a transplanted one.

Table 5: Guidance hyper-parameters. This single configuration is shared by every MAG result in the paper, across all three backbones and all seven guided attributes.

parameter	value	role
η (step size)	0.1	inner-loop step size
t_{start} (delayed start)	500	guide only once $t < 500$
$\gamma_{\max}$ (gradient clamp)	2.0	per-sample gradient-norm limit
K (DCT band limit)	12	temporal bases kept by Π_K

All backbones are MDM (Tevet et al., 2023) transformer encoders: 8 layers, latent width 512, DDPM 1000 steps (Ho et al., 2020), CLIP-ViT-B/32 text encoder. Each can be trained on a single NVIDIA A10 GPU (batch 32, lr 10^{-4}): the HumanML3D vanilla and AttrCond models for 600k steps, the KIT-ML and Basketball vanilla models for 200k steps. In AttrCond, the z-scored 9-dim attribute vector and its applicability mask are concatenated to the CLIP embedding and projected back to the latent width by a linear layer, so training starts exactly at the vanilla checkpoint.

D Off-Target Leakage

In this section, we check whether a single command drags the other attributes with it. Every clip in the MAG sweep is scored on all eight attributes, giving the matrix $A_{ij} = \rho(\text{commanded}_i, \text{realized}_j)$ in Table 6, whose diagonal entries measure the control response and whose off-diagonal entries measure leakage. Three groups account for almost all of the leakage, and we take them in turn before summarising the remainder.

Table 6: MAG off-target leakage on HumanML3D. Rows are the commanded attribute, columns the measured one, entries Pearson ρ, computed on the same runs as Table 2. Because the matrix must place every attribute on one scale, facing change enters here as the recovered angle $\text{atan2}(\sin\Delta\psi, \cos\Delta\psi)$ in degrees, whereas Table 2 correlates its sin–cos encoding as the guidance objective does, this is why its diagonal entry reads 0.34 here against 0.63 there, two statistics of the same runs rather than a discrepancy. The diagonal averages 0.70, rising to 0.86 if facing change and tempo are set aside. Leakage concentrates in three places: the kinematic identities among speed, path length and net displacement, the mutual coupling within the vertical group, and the tempo column. All entries outside these three groups satisfy $|\rho| \leq 0.27$.

Target \ Measured	Speed	Path	Disp.	Facing	Jump	Crouch	Reach	Tempo
Speed	**0.92**	0.92	0.78	-0.05	-0.03	-0.01	0.04	-0.52
Path length	0.97	**0.97**	0.82	-0.14	-0.14	-0.15	-0.14	-0.27
Net displacement	0.90	0.90	**0.97**	0.01	-0.03	-0.06	0.00	0.27
Facing change	-0.03	-0.04	-0.20	**0.34**	0.15	0.07	0.00	-0.25
Jump height	0.27	0.27	0.16	-0.13	**0.76**	-0.34	0.09	-0.06
Crouch depth	0.14	0.15	0.17	0.22	-0.28	**0.54**	-0.38	0.11
Reach height	0.00	0.11	-0.04	0.20	0.84	-0.40	**0.98**	-0.07
Tempo	-0.00	-0.00	-0.08	0.12	-0.10	-0.05	0.00	**0.10**

Kinematic identities. The entries of 0.78–0.97 among speed, path length and net displacement follow from $\text{path} \approx \text{speed} \times T/\phi$ on fixed-length clips, together with the fact that displacement couples to both under near-straight paths. Any control method co-moves these three by construction, so this block is a property of the attributes rather than of the interface.

Coupling within the vertical group. Jump, crouch and reach share the same vertical degrees of freedom and couple accordingly. The strongest instance is a reach command raising jump height (0.84), and it is asymmetric: the reverse direction is clean (0.09). Raising the hands is naturally

accompanied by ankle extension or a small hop, whereas jumping does not require the hands to go up. The negative entries in the crouch column (-0.34 from jump, -0.40 from reach) follow the same logic in the opposite direction, since an upward target suppresses how low the hip descends.

The Tempo. Tempo is the leakiest column in the matrix, with $|\rho|$ up to 0.52 from speed and 0.25–0.27 from path length, net displacement and facing change. We read this as a property of the tempo estimator rather than a kinematic anti-correlation: foot-contact detection thins out at high speed, and the rising-edge count is coarse on short clips. Tempo is also the attribute with the weakest diagonal response (0.10), so its column and its diagonal point to the same underlying measurement issue.

Separation elsewhere. Outside these three groups leakage is small, though not perfectly symmetric. Commanding speed leaves every vertical attribute within $|\rho| \leq 0.04$, while the reverse direction is weaker but nonzero, with jump reaching 0.27 on speed. Facing change is the one attribute that neither leaks into nor receives leakage from any group beyond 0.22. Three columns are not dominated by their own diagonal, namely speed (path reaches 0.97 against 0.92), jump (reach reaches 0.84 against 0.76) and tempo, and in each case the responsible entry belongs to one of the three groups above rather than to an unexplained interaction.

Per-level resolution. Table 7 resolves the three representative entries discussed above by commanded level, which makes visible what the aggregate correlations compress. Two readings follow. The reach→jump entry is driven almost entirely by the two stress commands in the upper tail, where realized jump rises to 0.30 and 0.63 m, whereas inside the feasible range it stays at or below 0.07 m: the coupling is a property of the stress regime, not of in-range control. The jump→crouch entries instead decline smoothly across the whole sweep, consistent with the estimator effect and the vertical trade-off described above.

Table 7: Per-level resolution of the three notable off-diagonal leakage entries (mean±std over 8 samples per level).

Reach→Jump (ρ=0.84)		Speed→Tempo (ρ=−0.52)		Jump→Crouch (ρ=−0.34)	
Reach	Jump	Speed	Tempo	Jump	Crouch
0.07	0.067±0.030	0.09	2.42±0.53	0.02	0.391±0.206
0.13	0.017±0.029	0.18	1.83±0.86	0.04	0.254±0.106
0.22	0.000±0.000	0.30	1.77±0.34	0.07	0.237±0.055
0.35	0.000±0.000	0.39	1.79±0.56	0.10	0.201±0.066
0.47	0.056±0.119	0.48	1.75±0.59	0.14	0.196±0.052
0.61	0.016±0.028	0.61	2.04±0.68	0.24	0.158±0.061
0.75	0.058±0.047	0.73	1.71±0.73	0.35	0.148±0.039
1.29	0.297±0.157	2.85	1.33±0.46	0.75	0.140±0.061
1.83	0.625±0.149	4.98	0.75±0.66	1.16	0.169±0.074

E Multi-Attribute Control under Joint Commands

We command three attribute pairs on 3×3 grids, taking the p_{20}, p_{50} and p_{80} level of each attribute from its single-attribute sweep, with 8 samples per cell and a caption compatible with both attributes. This experiment uses the same guidance configuration as the main results. Table 8 compares each attribute under a joint command against its single-attribute sweep, and Table 9 gives the nine cells behind each row, with row and column margins that isolate one attribute at a time. Since the three levels span only p_{20} to p_{80}, whereas the single sweeps span nine levels across the full feasible range, ρ and P_{50} are both deflated by the narrower command range, and α is the column that transfers most directly between the two settings.

Attributes realized at a single keyframe retain their control under a simultaneous second command. In the Jump+Reach grid both follow their own command: realized jump averages 0.11, 0.19 and 0.31 m against commands of 0.07, 0.14 and 0.35 m, and realized reach averages 0.24, 0.45 and 0.63 m against 0.22, 0.47 and 0.75 m. This is the pair with the largest off-diagonal entry in the leakage matrix (reach→jump $= 0.84$, Table 6), so the co-movement observed when jump is left unconstrained is a tendency the guidance can override once jump is commanded explicitly, rather than a constraint it must obey. Jump likewise holds its precision when paired with speed.

Attributes obtained by integrating the root trajectory do not compose, and they fail in two distinguishable ways. Paired with jump, commanded speed keeps its ordering (1.07, 1.70 and 2.00 m/s) but overshoots every command by roughly 1 m/s, so the α of 2.09 reflects a gain error rather than a loss of control. The compound caption's locomotion prior sets the pace: across the eighteen speed cells, "jogs" realizes anywhere from 0.77 to 2.75 m/s. Paired with facing, the ordering itself is lost (1.49, 1.20 and 1.38 m/s, $\rho = -0.04$), and facing is compressed to roughly a fifth of its commanded amplitude (-15, -2 and $17°$ for commands of -90, 0 and $90°$) with a per-cell spread of up to $\pm 110°$. Forward progression and turning contend for the same planar root trajectory within a fixed 60-frame clip, so neither command can be met without giving ground on the other.

The boundary between the two classes is sharp and follows from how each attribute is realized. A jump and a reach occupy different joints at possibly different frames, and can be satisfied independently. A speed and a turn are both integrals of one planar trajectory of fixed duration, and a single trajectory cannot satisfy two such budgets at once. Users issuing joint commands should therefore expect pose-level attributes to compose and trajectory-level attributes to require a consistency check, which we discuss further in Appendix J.

Table 8: Dual-attribute control against single-attribute sweeps, reported as α / ρ / P_{50} after marginalizing over the second commanded attribute.

Pair	Attribute	Joint command	Single sweep (Tab. 2)
Speed + Jump	Speed	2.09 / 0.23 / 0.41	0.79 / 0.92 / 0.49
	Jump	0.67 / 0.86 / 0.028	0.58 / 0.76 / 0.037
Speed + Facing	Speed	-0.20 / -0.04 / 0.71	0.79 / 0.92 / 0.49
	Facing	-0.05 / -0.06 / 83.8	0.40 / 0.63 / 45.6
Jump + Reach	Jump	0.67 / 0.60 / 0.042	0.58 / 0.76 / 0.037
	Reach	0.74 / 0.77 / 0.076	0.89 / 0.98 / 0.018

Table 9: The 3×3 grids behind Table 8. Rows are the commanded level of the first attribute, columns that of the second, and each cell gives realized$_1$ / realized$_2$ averaged over 8 samples. The margins isolate one attribute at a time: the rightmost column is the mean realized value of the row attribute and should follow the row labels, while the bottom row of each block is the mean realized value of the column attribute and should follow the column labels.

Speed + Jump	*"A person jogs forward and performs a jump."*			
Speed \ Jump	0.07	0.14	0.35	row mean
0.30	1.01 / 0.13	1.28 / 0.14	0.91 / 0.30	1.07
0.48	1.45 / 0.12	1.83 / 0.15	1.81 / 0.29	1.70
0.73	2.75 / 0.09	1.61 / 0.16	1.65 / 0.31	2.00
column mean	0.11	0.15	0.30	
Speed + Facing	*"A person jogs forward and turns."*			
Speed \ Facing	-90	0	90	row mean
0.30	1.21 / -29	1.47 / 16	1.78 / 38	1.49
0.48	1.10 / 28	0.77 / 1	1.72 / -15	1.20
0.73	1.55 / -43	1.48 / -22	1.10 / 29	1.38
column mean	-15	-2	17	
Jump + Reach	*"A person jumps and reaches upward."*			
Jump \ Reach	0.22	0.47	0.75	row mean
0.07	0.10 / 0.26	0.14 / 0.33	0.10 / 0.63	0.11
0.14	0.16 / 0.23	0.25 / 0.53	0.17 / 0.71	0.19
0.35	0.32 / 0.22	0.37 / 0.50	0.25 / 0.56	0.31
column mean	0.24	0.45	0.63	

F ADDITIONAL RESULTS ON HUMANML3D

Vanilla-MDM baselines. Table 10 reports the two vanilla baselines of §4.2 side by side. The left block measures *physical fidelity*: for each test caption we generate one clip and compare its attribute value against that of the corresponding reference clip, so ρ is a correlation between generated and reference values and P_{50} is the median absolute difference between them. The right block measures *numeric prompting*: the target value is written into the caption and the clip is scored by the control-curve protocol, so ρ, P_{50} and the α in this block relate the commanded value to the realized one. The two blocks therefore answer different questions. The left asks whether an unprompted sample lands near the reference, and the right asks whether a number in the caption is obeyed. Numeric prompts use one template per attribute, of the form *"jumps exactly v meters high"*. Facing change is encoded as a sin–cos pair, which the two blocks resolve differently: the left block measures each channel separately, whereas the right block commands a single angle and reports one entry spanning both rows, with ρ computed on the sin–cos encoding and P_{50} as a circular difference in degrees, as in Table 2.

Table 10: Vanilla-MDM baselines on HumanML3D. Left: physical fidelity on the test set of $4{,}384$ clips, with N the number applicable to each attribute. Right: the same backbone under numeric prompting, scored by the control-curve protocol. Units of P_{50} are $\mathrm{m/s}$ for speed, deg for facing change, Hz for tempo, and m for the remaining physical attributes. The sin and cos channels are dimensionless.

	Physical fidelity			With numeric prompting		
Attribute	N	ρ	P_{50}	α	ρ	P_{50}
Speed	2237	0.424	0.518	-0.02	-0.03	1.42
Path length	4012	0.488	1.029	0.01	0.07	2.01
Net displacement	2709	0.336	0.732	0.03	0.10	1.67
Facing change (sin)	4384	0.061	0.218	0.03	0.32	52.4
Facing change (cos)	4384	0.393	0.033			
Jump height	1272	0.384	0.091	-0.02	-0.12	0.23
Crouch depth	1709	0.219	0.076	-0.15	-0.20	0.27
Reach height	3537	0.484	0.179	-0.00	-0.00	0.35
Tempo	1560	-0.135	1.156	-0.00	-0.01	1.40
mean (exclude Tempo)	—	**0.349**	—	≈ 0	**0.02**	—

The two blocks separate two failure modes that are easily conflated. Physical fidelity is moderate, averaging $\rho = 0.349$, which is what one expects from a model trained on captions that describe actions rather than magnitudes. It produces plausible walking, but not walking at a specified speed. Numeric prompting collapses to $\rho = 0.02$ with slopes indistinguishable from zero on every attribute, so the shortfall is not that the backbone lacks the capacity for graded motion, but that the caption is not a channel through which a magnitude can enter.

Why we do not recalibrate the command. The control-curve slopes are well below one on the trajectory aggregates, and a fitted slope invites an obvious remedy: fit α and β once per attribute from a calibration sweep, then send $c' = (c - \beta)/\alpha$ instead of c. If the response were exactly affine this would set the slope to one at no cost. Table 11 reports what such a correction could achieve, by decomposing the current error into the part a fitted line explains and the part it does not: the residual column is the median deviation from the attribute's own fitted line, and is the floor a perfect recalibration would leave.

The decomposition splits the interface in two kinds. On the three trajectory aggregates most of the error is the systematic gain term, and removing it would reduce the median error by a factor of 1.6 to 2.5. On the three vertical extrema the residual is as large as the current error or larger, so there is nothing for a gain correction to remove, their errors are scatter around a line that already passes near the target, which is also why their P_{50} is already at the centimetre scale.

We do not adopt the correction, because the affine model it assumes fails exactly where it would be needed. The lowest speed level asks for $0.089\,\mathrm{m/s}$ but realizes $0.848 \pm 0.272\,\mathrm{m/s}$, and no sample falls below $0.568\,\mathrm{m/s}$: the caption describes a person walking, so the backbone will not produce a near-stationary clip under it. The fit absorbs this floor into its intercept of $0.708\,\mathrm{m/s}$. Inverting it at

this level asks for -0.78 m/s, but the negative value is only a symptom; the real obstacle is that no command, however small, reaches below the floor the caption imposes. Part of what the fit reports as a gain is that floor, and rescaling the command cannot move it. Only a different caption can. Additionally, two further caveats apply to the residual column. It is computed on the samples the current command produced, so treating it as the error that would remain after recalibration assumes the residual is unchanged when the command moves, which we have not verified. And α and β are fitted per attribute on one caption; whether a single pair transfers across captions is exactly what the speed floor calls into question. We therefore report the response as measured and leave the correction to a caller who can calibrate it for their own prompt distribution.

Table 11: What an affine recalibration of the command could achieve, on the HumanML3D sweeps of Table 2. α and β are the control-curve fit $r = \alpha c + \beta$. *Current* is the median $|c - r|$ as reported in Table 2; *residual* is the median $|r - (\alpha c + \beta)|$, the scatter around the fitted line, which a gain correction cannot remove and which therefore bounds what recalibration could deliver. Units are the attribute's own.

Attribute	α	β	current P_{50}	residual
Speed	0.79	0.708	0.49	0.30
Path length	0.42	0.937	1.34	0.53
Net displacement	0.52	0.468	0.60	0.29
Jump height	0.58	0.093	0.037	0.044
Crouch depth	0.49	0.138	0.102	0.157
Reach height	0.89	0.046	0.018	0.024

Text fidelity under commands that carry no reference information. Table 1 commands every clip its own ground-truth attribute vector. That is the natural choice for a retrieval metric, since it keeps the command inside what the caption describes, but it also means the command is measured from the very motion the retrieval is scored against, and a sceptical reading would attribute part of MAG's margin to that. We therefore repeat the measurement with two command sources that carry no information about the clip being generated, with the vanilla backbone for reference (Table 12).

Retrieval does not depend on the command being the clip's own. Commands drawn from the sweep grid give $R@1 = 0.461$ and commands from an unrelated clip give 0.464, against the vanilla model's 0.450. All three conditions lie within 0.02 of one another on $R@1$ and MM-Dist. That last gap, 0.469 against 0.461 and 0.464, is the part of MAG's margin in Table 1 that comes from the command agreeing with the caption, since a clip's own attributes are by construction consistent with the caption that describes it. This rules out one alternative reading of Table 1, that retrieval improves because guidance pulls the sample towards the reference motion the retrieval is scored against. Commands that carry no information about that reference leave retrieval no lower than the unguided backbone, which is what preserved text fidelity requires.

Table 12: Text fidelity of MAG under three command sources, on the full HumanML3D test split, under the shared guidance configuration of Appendix C. *Ground truth* commands each clip its attributes as measured from its own reference motion, as in Table 1. *Sweep grid* draws each commanded attribute uniformly from the nine percentile levels. *Permuted* assigns each clip the attribute vector of another clip, preserving the marginal distribution while removing any correspondence to the caption. The vanilla row takes no attributes and is unaffected by the choice.

Model	Command source	R@1↑	R@3↑	MM-Dist↓
Vanilla MDM	–	0.450	0.713	1.018
MAG	Ground Truth (Table 1)	0.469	0.727	1.004
	Permuted across clips	0.464	0.723	1.006
	Sweep grid	0.461	0.720	1.008

The commanded levels and what they span. A command is feasible where the dataset support and the caption's semantic support overlap, and this part characterises both. The nine commanded levels of every sweep are placed from the marginal distribution of the attribute over the training split, at $0.3\,p_{20}$, $0.6\,p_{20}$, p_{20}, $(p_{20}{+}p_{50})/2$, p_{50}, $(p_{50}{+}p_{80})/2$, p_{80}, $(p_{80}{+}\max)/2$ and $\max$. Levels three

through seven therefore lie within $[p_{20}, p_{80}]$, the central 60% of the observed values, while the first two probe below the twentieth percentile and the last two above the eightieth, the ninth being the largest value the split contains. We refer to the last two as *stress levels*: they are inside the empirical support but in its sparsely populated tail, which is where we expect a frozen backbone to run out of examples to draw on. All experiments are on the frozen HumanML3D backbone of §4.3. Table 13 gives the median error at each level separately, which is the resolution the question needs. Reporting a slope or a correlation over a subset of levels would not be informative, because restricting the command range shrinks the signal while leaving the per-level spread untouched: within $[p_{20}, p_{80}]$ the speed command varies by only 0.44 m/s whereas realized values scatter by 0.2 to 0.6 m/s per level, so a α fitted there measures noise. Error at a fixed command does not have this problem.

A separate measurement asks whether accuracy and physical validity break at the same point. On the reach sweep, where the bone chain at the realization frame can be tracked level by level, the mean chain length stays within $\pm 3.6\%$ of the clip median at every level up to the eightieth percentile, and it is the two levels beyond it that both miss the command and deform the chain, by $+17.9\%$ and $+24.6\%$ on average. The jump sweep behaves the same way: chains hold to within $+4.4\%$ through the levels the caption supports, and the level that overshoots is the level that stretches them, by $+21.3\%$ at its worst sample. Where the error is small the motion is therefore physically clean, and the two failures coincide rather than accumulating separately. These checks cover the three vertical extrema, whose realization is localized to a key frame, we have no comparable per-level check for the trajectory aggregates, whose realization is spread over the whole clip.

The per-level view shows a flat interior and a tail that fails, not a gradual decay. Across $[p_{20}, p_{80}]$ the median error is close to constant, and for jump height it *falls* as the command grows, from 0.043 to 0.021 m, since the caption already produces a jump of roughly the middle magnitude. Both attributes whose P_{50} we quote as centimetre-level in the main text stay there across the whole interior: at most 0.043 m for jump and 0.044 m for reach. The failure is concentrated at the ninth level, and it is abrupt rather than progressive: jump error jumps from 0.030 to 0.413 m between the eighth and ninth level, and path length from 7.9 to 12.0 m, while the eighth level is still within a factor of 1.5 of the interior for four of the six attributes. This is the behaviour that motivates treating the extreme of the support as a separate regime, and the chain-stretch measurements below show what the model does there instead of following the command. Facing change and tempo are omitted: the facing sweep spans the full angular circle, so its outermost levels at $\pm 180^\circ$ are a wrap boundary rather than a tail, and tempo is not guided.

Table 13: Median absolute error $|c - r|$ at each commanded level, in the attribute's own unit, on the runs of Table 2. Levels are ordered by magnitude, upright entries lie within $[p_{20}, p_{80}]$ of the training distribution and *italic* entries outside it, the last column being the largest value in the split. Error is flat across the interior and breaks down at the final level.

	below p_{20}		within $[p_{20}, p_{80}]$					above p_{80}	
Attribute	1	2	3	4	5	6	7	8	9
Speed	*0.68*	*0.65*	0.74	0.44	0.27	0.34	0.30	*0.51*	*0.17*
Path length	*1.39*	*1.25*	0.72	0.43	0.45	0.80	1.74	*7.94*	*12.04*
Net displacement	*0.52*	*0.36*	0.48	0.40	0.17	0.39	1.18	*2.65*	*4.47*
Jump height	*0.054*	*0.047*	0.043	0.043	0.033	0.023	0.021	*0.030*	*0.413*
Crouch depth	*0.019*	*0.039*	0.064	0.084	0.149	0.079	0.133	*0.144*	*0.458*
Reach height	*0.019*	*0.015*	0.016	0.018	0.022	0.009	0.044	*0.020*	*0.120*

The marginal distribution is also an upper bound on what a given caption admits. Because the levels are placed by pooling all captions, they do not account for the one a clip is generated from. To locate the caption-bounded boundary we swept additional levels under the same seed and configuration. For the jump caption it lies just above the range of Figure 3. At $c = 0.50$ m the level statistics turn bimodal: three samples undershoot at 0.09 to 0.19 m and four reach 0.49 to 0.52 m, giving a level mean of 0.33 ± 0.16 m. The model does not interpolate towards an unreachable target, it either attains it or abandons it. At $c = 0.70$ m, a value the training split does contain and which sits below its eighth commanded level, the samples that do hit the target reach it through leg-chain stretch of 11 to 47%. Both supports therefore constrain the command, and at this level the caption is the tighter of the two.

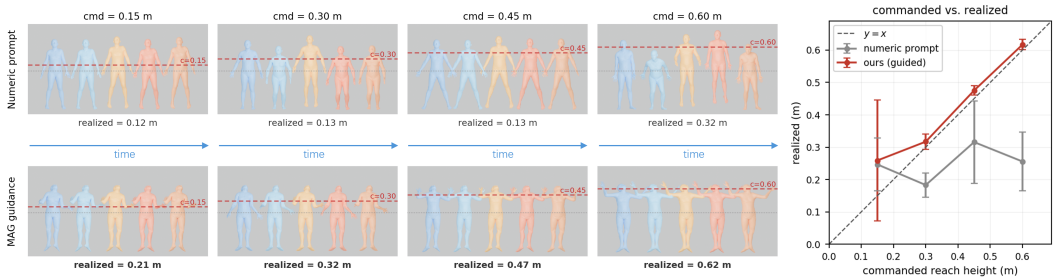


Figure 4: **Reach-height sweep on the frozen HumanML3D backbone** (caption *"A person extends their arms upward."*, commanded 0.15–0.60 m). Each cell shows the median-realized sample. Annotations give mean±std over 8 samples.

Seed robustness of the main sweep. We repeated the jump-height sweep of Figure 3 with two further seeds under an otherwise identical configuration. Both the guided and the prompted curve behave as in the figure: realized height rises with the command under guidance, with per-seed slopes of 0.55–0.78 and ρ of 0.84–1.00, and stays flat under numeric prompting. Agreement between seeds is tight at the lower commands, where level means fall within 0.01–0.03 m of one another for $c \leq 0.20$ m, and loosens to 0.07–0.11 m at $c = 0.30$ and 0.40 m, which is the first indication that the upper end of the sweep approaches the caption's limit. Pooling the three runs gives 24 samples per level, with means and standard deviations of 0.13 ± 0.05, 0.21 ± 0.05, 0.29 ± 0.08 and 0.33 ± 0.10 m for commands of 0.10, 0.20, 0.30 and 0.40 m. The first three track the command closely; the last falls short of it by 0.07 m, a gap examined below.

Reach-height sweep. Figure 4 repeats the main-paper protocol (Figure 3) on the same frozen HumanML3D checkpoint with a different attribute and caption (*"A person extends their arms upward."*, commanded reach height 0.15–0.60 m). Tracking is even tighter than for jump height: the guided rows lock onto the commanded line while the numeric prompt stays flat. Guided samples keep both feet grounded, the standing-reach caption is honored while the scalar command is met.

G Additional Internal Validation: Basketball Testbed

Dataset. The Basketball testbed is an internally maintained mocap corpus of 20,286 single-person basketball clips at 30 fps, spanning 8 categories (shot, layup, dunk, pass, block, rebound, catch, and locomotion). Each clip is paired with captions and, uniquely, with an independently produced per-subtype numeric label vector that is distinct from the extractor f used throughout the paper. The dataset is proprietary, so the numbers below serve purely as internal validation. Because Basketball dataset has GT numeric labels that no public T2M dataset carries, it enables two independent validations that the public benchmarks do not support: (i) calibration of the extractor f against labels produced without any reference to f; and (ii) an in-the-wild sanity check where the caption style, action distribution, and frame rate all differ from HumanML3D and KIT-ML.

Running f on GT clips and comparing against the independent numeric labels grounds the measurement instrument used throughout the paper: on every field family that this paper commands, the extractor agrees with the labels at a median $\rho \geq 0.96$, and the vertical extrema calibrate exactly at $\rho = 1.000$, $P_{50} = 0$.

In-domain guidance. We train a third vanilla MDM on the Basketball dataset, using the same architecture and hyper-parameters as for HumanML3D and KIT-ML, with only the frame rate following the data, and apply MAG without changing any guidance hyper-parameter. Table 14 reports the control curves. The vertical attributes, for which the extractor is exactly calibrated against per-frame GT, reach $\rho = 0.95$ to 1.00, with P_{50} of 3.3, 1.9 and 5.7 cm for jump, crouch and reach against a GT floor of 0 cm. Horizontal aggregates are controlled in direction but saturate in range, and facing change fails outright, since Basketball clips are short and largely in place. This follows the manifold rule of §4.4: guidance steers within the manifold the backbone has learned, and the Basketball distribution contains too few turning motions for a facing command to act on.

Table 14: MAG control curves on the Basketball-trained backbone (in-domain, 9 levels $\times$ 8 samples per attribute; α / ρ / P_{50}). Same hyper-parameters as HumanML3D and KIT-ML.

Attribute	α	ρ	P_{50}
Speed	0.45	0.76	1.41
Path length	0.32	0.87	1.62
Net displacement	0.52	0.94	2.93
Facing change	-0.15	-0.09	99.1
Jump height	0.91	1.00	0.033
Crouch depth	0.85	0.95	0.019
Reach height	0.91	1.00	0.057
Tempo	0.06	0.07	0.87

Qualitative results. Figure 5 repeats the sweep of Figure 3 on the Basketball-trained backbone, where per-clip GT labels calibrate the measurement to a 0 cm floor. The guidance configuration is unchanged ($\eta = 0.1$, $t_{\text{start}} = 500$, $\gamma_{\max} = 2$, $K = 12$), with only the frame rate following the data, and the commanded levels are drawn from the feasible range of §4.3. Within the feasible range, realized reach height tracks the command closely: $0.40 \to 0.480 \pm 0.037$, $0.60 \to 0.650 \pm 0.033$ and $0.80 \to 0.823 \pm 0.045$ m. The arm chain is not recruited to achieve this, its mean length at the reach apex staying between -1.3% and -4.1% of the clip median across these levels, with a worst single sample of $+14.7\%$ at 0.80 m. This is the qualitative counterpart of the $\rho = 0.95$ to 1.00 rows of Table 14. Beyond that range the numbers remain accurate but the motions do not. Commands of 1.00 to 1.60 m exceed what a standing reach can produce, yet are still met to within 2 cm ($1.00 \to 1.016 \pm 0.048$, $1.20 \to 1.203 \pm 0.045$, $1.60 \to 1.579 \pm 0.012$ m). They are met by stretching the arm chain, whose mean elongation grows from $+2.6\%$ at 1.00 m (maximum $+27.0\%$) through $+7.6\%$ at 1.20 m (maximum $+39.3\%$) to $+34.9\%$ at 1.60 m (maximum $+67.0\%$), and in part by an involuntary hop, with the apex root rising 0.27 m at $c = 1.00$ m against 0.10 to 0.13 m at every feasible level.

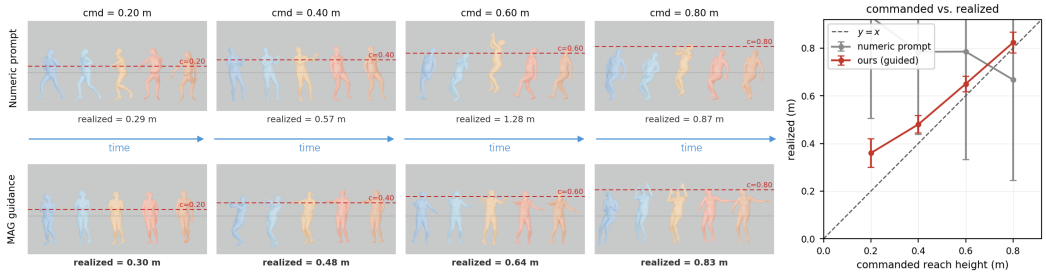


Figure 5: **Reach-height sweep on the Basketball-trained backbone.** Caption: *"A person extends their arms upward."* Layout follows Figure 3: each cell shows the median-realized sample, annotations give mean $\pm$ standard deviation over 8 samples, and the right panel plots commanded against realized height.

H OmniControl: Interface Boundaries of Geometric Control

This part expands the interface-class analysis of OmniControl summarized in §4.3. Commanding an attribute through the (joint, frame, xyz) interface requires converting a scalar target into a set of coordinate constraints. Whether this conversion is possible depends on how the attribute is defined over the clip, and the eight attributes fall into three classes.

Attributes that coincide with the interface. Net displacement is the pelvis position at the final frame, and speed determines a dense per-frame root trajectory. Both translate into coordinate constraints without loss. OmniControl attains $1.00/1.00/0.00$ on net displacement, where a single coordinate triple carries the whole command, and $0.79/0.99/0.04$ on speed, where the command becomes T constraints and the residual error is the model's tracking of them. The correspondence is exact rather than learned: the command and the interface input are the same quantity, so the re-

sult measures the fidelity with which the model reproduces a specified trajectory, not its ability to translate a scalar into one.

Clip-level extrema. Jump, crouch and reach heights are extrema over all frames of a clip, and a finite set of equality constraints determines such an extremum only from below. Pinning one frame high raises the extremum to at least that height, but pinning one frame low leaves the model free to attain a larger value at any unconstrained frame. The response is therefore one-sided. For jump height, commands from 0.02 to 0.35 m all yield a realized value near 0.40 m, the height the caption produces on its own, and the curve only begins to follow the command above that point. All of the reported P_{50} error originates in this flat half-domain, which we mark with * in Table 2. Bounding an extremum from above would require inequality constraints on all T frames, which the equality-based interface does not provide.

Attributes with no native interface. The remaining three have no scalar entry point, though they differ in why. Tempo is a property of the contact signal in the frequency domain, and we are not aware of a reduction of it to a finite set of coordinate constraints. Path length and facing change are reachable in principle but only by over-specifying: on fixed-length clips path length is proportional to mean speed, so the dense-trajectory construction we use for speed would also realize a commanded length, and a heading difference can be imposed by placing the hip joints at both ends of the clip. In each case the user must supply a full trajectory or a pair of joint placements, which fixes the shape of the motion as well as the scalar and therefore answers a different question than the one the command asked. We therefore report these as outside the *scalar* interface rather than geometrically impossible.

These three classes describe a boundary rather than a deficiency. Geometric control specifies where individual joints must be, and is the appropriate interface when that is what the user knows. A scalar attribute specifies a property of the clip as a whole, and the two are largely complementary: the extremum and out-of-interface attributes are reachable by an optimizer that descends the attribute function directly, which is what MAG and DNO do and what the geometric interface cannot express.

I DNO Baseline: Setup, Physical Quality, and the Π_K Transplant

Setup. DNO (Karunratanakul et al., 2024) optimizes the initial diffusion noise z by backpropagating a motion-space loss through the entire denoising chain. We run the official optimizer (Adam, learning rate 5×10^{-2}, warm-up followed by cosine decay, gradient normalization, noise decorrelation) on our frozen HumanML3D checkpoint, with a differentiable 100-step DDIM unroll under gradient checkpointing and our attribute loss (Eq. 3) as the objective. The loss is the one MAG descends, with none of MAG's gradient-side machinery, so the two mechanisms are compared on a shared objective. Optimization runs for at most 300 steps under the same early-stop criterion as MAG: mean absolute z-score error below 0.05, checked every 10 steps. On single-attribute sweeps DNO terminates in 29 to 103 steps. On the eight-attribute R-Precision protocol (Table 1) it averages 147 steps, reaches the 300-step cap on 37% of batches, and stalls at 0.06 to 0.14 z-score error, so satisfying several competing attributes at once is harder in noise space than satisfying one.

We run DNO with a 100-step DDIM unroll, which is the setting its official implementation uses for this purpose. The DNO paper also reports a 10-step configuration, chosen there as a compromise between quality and resource use for a different task, and we did not adopt it. Because the unroll length sets how faithfully the gradient reflects the sampler that will actually generate the clip, and a shorter unroll would reduce DNO's cost while also changing the objective it descends. Since our comparison is about where the update is placed rather than how cheaply either method can be run, we kept the official setting and instead report MAG at a matched 100-step sampler alongside its own DDPM-1000 figure, so that the cost columns can be read either way.

Accuracy and cost. Table 2 reports the seven differentiable attributes. DNO's per-level means track the commands closely (mean $\rho = 0.920$), as several hundred dedicated descent steps per clip should. The cost is 63.9 s per clip averaged over attributes on one A10 GPU. Under a matched 100-step sampler MAG takes 2.79 s per clip, and 8.80 s at DDPM-1000 (jump sweep, identical batch). MAG's control is insensitive to the sampler: on the 100-step sampler its jump curve is essentially unchanged, with α, ρ and P_{50} of 0.66, 0.995 and 0.056 against 0.70, 0.995 and 0.058 at 1000 steps.



Figure 6: Visual evidence for the physical-quality gap at an in-range command (reach $c = 0.35$ m). Left: Five consecutive frames of DNO and MAG. For the DNO, at frame-28 the arm chain collapses to -29% of its median length for a single frame, then recovers. MAG at the same command, chain length stable throughout. Right: arm-chain length deviation over time (shading: $\pm 5\%$ noise floor). The transient collapse is invisible to whole-clip accuracy metrics, but is exactly what Jitter and Peak jerk (Table 2, Table 15) capture.

Physical quality. The two mechanisms separate on motion quality rather than on accuracy. DNO's per-clip optimization introduces transient kinematic glitches, including single-frame chain collapses of up to -29% at in-range reach commands (Figure 6). Whole-clip accuracy is blind to these, whereas standard smoothness metrics are not. Table 15 reports Jitter, Peak jerk and foot-skate ratio averaged over within-range levels, with unguided numeric-prompt sampling included as a reference point. The comparison splits by attribute class. On reach, where optimization pressure meets an essentially static pose, DNO's Jitter is 3.8 times MAG's and its peak jerk 5.2 times. On crouch the two are comparable. On jump both roughly double the unguided Jitter, since an impulsive take-off amplifies any guidance pressure. MAG's advantage is thus specific to sustained-pose attributes.

The band limit is what suppresses single-frame spikes, and the MAG (K=0) column isolates its effect: removing only Π_K raises Peak jerk from 221 to $3518\,\mathrm{m/s^3}$ on reach and from 486 to 2107 on crouch, while Jitter moves far less, from 0.91 to 1.29 and from 1.71 to 2.16. The damage is concentrated in isolated frames, which is what the projection removes. On jump the contrast disappears (5866 against 5344), consistent with the preceding paragraph. The effect is not tied to the vanilla backbone: AttrCond+G, which applies the same guidance configuration to the attribute-conditioned backbone, reaches comparable or lower peak jerk on all three attributes.

The baselines carry recognizable signatures. OmniControl shows the effect of localized spatial pulls: its Jitter is low on crouch and reach, but its Peak jerk is the highest of any method on those two attributes, and its foot-skate rate is the highest on jump. The jerk peak lands on the hinted wrist joint within two frames of the hint, as the constrained joint is drawn onto its target while the rest of the body glides. AttrCond attains the lowest Peak jerk throughout, but this follows from weak control (reach α 0.48), remaining close to the Unguided manifold is automatically smooth when the command is not enforced.

The Π_K transplant does not rescue DNO. To separate mechanism from regularizer we injected the band limit into DNO's objective as an identity-forward autograd operator that projects $\partial\mathcal{L}/\partial x$ onto the first K=12 DCT bases before it enters the unrolled denoiser. Accuracy is broadly preserved on two attributes (jump $0.52/0.940$; reach improves to $0.98/0.998$) but crouch control collapses to an $alpha$ of 0.03 and $\rho = 0.08$. And the transient glitches are not repaired, the maximum in-range chain deviation remains $+36\%$, at crouch $c = 0.08$ m. The projected gradient still passes through the nonlinear noise-to-motion map, which re-synthesizes high-frequency content from band-limited noise updates. The same projection works where MAG applies it, on the late-step posterior mean adjacent to the final sample, so it is the placement of the update rather than the frequency-domain regularizer that yields physically clean control.

Table 15: Physical-quality metrics on the extremum sweeps, averaged over within-feasible-range levels. Jitter is the mean joint acceleration magnitude (m/s^2), Peak jerk the maximum single-frame jerk magnitude (m/s^3), and Foot skate the fraction of ground-contact frames sliding more than 2.5 cm. *Unguided* denotes numeric-prompt sampling of the same backbone and serves as the noise floor. All columns are measured at identical commanded levels. OmniControl generates its fixed 196-frame length against 60 frames elsewhere. Jitter is a per-frame mean and comparable across lengths, but Peak jerk is a maximum over frames and is not: a longer sequence has more frames in which to attain it, so OmniControl's Peak jerk should be read as an upper bound rather than a matched comparison. Comparisons among the 60-frame columns are unaffected. AttrCond+G uses the shared guidance configuration of Appendix C, while MAG ($K{=}0$) differs from MAG in the band limit alone and therefore isolates the effect of Π_K.

Attribute	Metric	OmniCont.	DNO	AttrCond	AttrCond+G	MAG ($K{=}0$)	MAG	Unguided
Jump height	Jitter	6.76	28.1	5.30	5.31	23.3	24.9	13.4
	Peak jerk	2554	6726	779	777	5344	5866	2578
	Foot skate	0.71	0.58	0.15	0.15	0.52	0.57	0.25
Crouch depth	Jitter	0.74	2.09	0.84	0.84	2.16	1.71	1.95
	Peak jerk	5107	490	149	149	2107	486	554
	Foot skate	0.05	0.07	0.00	0.00	0.04	0.05	0.07
Reach height	Jitter	0.38	3.45	0.69	0.67	1.29	0.91	10.3
	Peak jerk	7072	1147	106	109	3518	221	1818
	Foot skate	0.13	0.01	0.00	0.00	0.00	0.00	0.16

J Limitations and Discussion

J.1 Discussion of Control mechanism

We treat the interior of the attribute's marginal distribution as the operating specification (§4.3) and release beyond-range clips as labelled stress material. That range is a statistic of the dataset and is blind to the caption, so it neither guarantees that an in-range command is physically realizable under a given prompt, nor that an out-of-range one is not. Probing intermediate levels locates caption-bounded boundaries that sit inside the percentile range: a jump command of 0.70 m, below the eightieth percentile of the split, is already realized in part through leg-chain stretch of 11 to 47%. The percentile range therefore selects where to look, and the direct measurement is what certifies an individual sample. Besides, guidance descends the soft attributes $\tilde{f}_\tau$ while evaluation measures the hard map f. The τ annealing shrinks this gap but does not close it, and the per-sample early-stop tolerance ($\varepsilon = 0.05$ in z-score units) bounds the precision that is achievable at all.

Controllability is bounded by the backbone's data distribution. Facing-change control fails on KIT-ML ($\rho \approx 0$) and Basketball, because those datasets contain few turning motions and the frozen backbone offers no manifold direction for guidance to exploit. Guidance redistributes probability mass within what the backbone learned, and it cannot synthesize behaviour the training distribution lacks. The same effect appears as saturation at extreme commanded values on all datasets, with control-curve slopes below one throughout.

Simultaneous control is bounded by competition for degrees of freedom. Multi-attribute commanding requires no additional machinery, since the guidance loss is a masked sum, and it preserves single-attribute precision for pairs that actuate different degrees of freedom. This holds even for jump and reach, which co-move when only one is commanded but separate once both are commanded explicitly. Pairs competing for the same trajectory degree of freedom behave differently. Commanding speed and facing change together forces forward progression and turning to share the root's planar path within a fixed clip length, and facing control drops to $\rho = -0.06$ from 0.63 when commanded alone, with a per-cell spread of up to $\pm 110^\circ$ (Table 8). The interface exposes eight scalars, but their joint feasible set is determined by kinematics rather than by the interface. A longer clip, or a temporally segmented schedule such as "turn, then jog", would relax the constraint, and we leave this to future work.

Guidance interacts with caption semantics only through the delayed start. R-Precision is preserved at the operating point we use, but we offer no formal guarantee, and a command far outside the cap-

tion's semantic range, for instance "walks slowly" with speed $= 4$ m/s, places the text and attribute terms in direct conflict. In that regime the outcome is settled by whichever signal dominates late denoising, and we do not control which.

Tempo is not differentiable, because its gait-cycle count contains a discrete edge detector. MAG therefore guides seven of the eight attributes and leaves tempo to the caption, and its near-zero control correlation ($\rho \approx 0.1$) follows from that design boundary rather than from a failure of the guided attributes. A smooth spectral surrogate, such as a soft peak of the foot-velocity spectrum, would be a natural extension.

J.2 Limits of the evaluation

Every number in our protocol passes through the attribute map f. It defines the command, guidance descends it, and the commanded–realized error measures it. On HumanML3D and KIT-ML the evaluation is therefore *method-self-consistent* rather than grounded in independently annotated physical labels, which no public T2M dataset carries. Where f determines the motion this is harmless, but f constrains joint positions and not bone lengths, so a commanded reach height can be met either by raising the arm or by lengthening it, and the protocol scores both as successes.

The two routes separate near the edge of the training support. On the HumanML3D reach sweep we measured the shoulder–elbow–wrist chain length at the reach apex, relative to the clip's own median. Across the interior of the distribution it stays within $+3.6\%$ of rest, while the two levels above p_{80} stretch it by $+17.9\%$ and $+24.6\%$ on average, with maxima of $+25.0\%$ and $+37.6\%$, giving visibly elongated arms. The eighth level is the informative case: it is accurate to 0.020 m, within the spread of the interior levels, and already stretches the chain by $+17.9\%$. The command is met, and met by deforming the body. The same pattern appears on the Basketball reach sweep, and the HumanML3D jump sweep shows the analogous root artifact, with part of the commanded apex realized by translating the root upward rather than by a ballistic jump.

No metric in the standard suite separates the two routes. R-Precision and FID, computed on the same evaluator features, measure neither smoothness nor physical plausibility. Here FID is the Fréchet distance in the evaluator's motion-feature space ($n = 1024$ samples per configuration, ground-truth split-half floor 0.008); it is flat across guidance settings, at 0.063 for vanilla MDM, 0.062 for MAG and 0.063 for the collapsed configuration, and it is not ordered by controllability, since the lowest value in our comparison, 0.053, belongs to AttrCond+G, which leaves facing change largely uncontrolled ($\rho = 0.18$) and which sits below both MAG and the backbone it guides. This mirrors on the evaluation side the observation that motivates our work, namely that text-trained systems treat a specification semantically rather than metrically, and it is why the commanded–realized protocol is needed alongside the standard metrics. That protocol does register the coarser failure of manifold collapse, giving a minimum ρ of 0.14 for the collapsed configuration, but as the eighth reach level shows, it does not register skeletal deformation.

We therefore rely on two measurements that lie outside f. The first is the chain length above, applied to the three vertical attributes, where a single chain and a single key frame make it well defined; the trajectory aggregates, whose realization is spread over the whole clip, have no comparable check, so their numbers rest on the percentile screen alone. The second is Basketball, whose independently produced numeric labels validate the extractor itself (median $\rho \geq 0.96$ on every commanded field family) and make its control results ground-truth-grounded, showing the same control behaviour there (jump-height $P_{50} = 3.3$ cm in-domain against a calibration floor of 0 cm). The physical-quality metrics we report alongside them, namely jitter, peak jerk and foot-skate ratio (Table 15), close part of the remaining gap by exposing the transient glitches that accuracy and FID both miss, but they remain hand-designed statistics. Two extensions would reduce the reliance on external checks: a bone-length consistency term in the guidance loss, which would prevent the deformation rather than detect it, and a learned plausibility metric predicting human judgements of motion realism, which would complete the suite. Both are left for future work.