TACD: DISTILLING EFFICIENT TEXT-TO-MOTION MODELS VIA TERMINAL AMPLIFICATION CONTROL

Anonymous authors
Paper under double-blind review

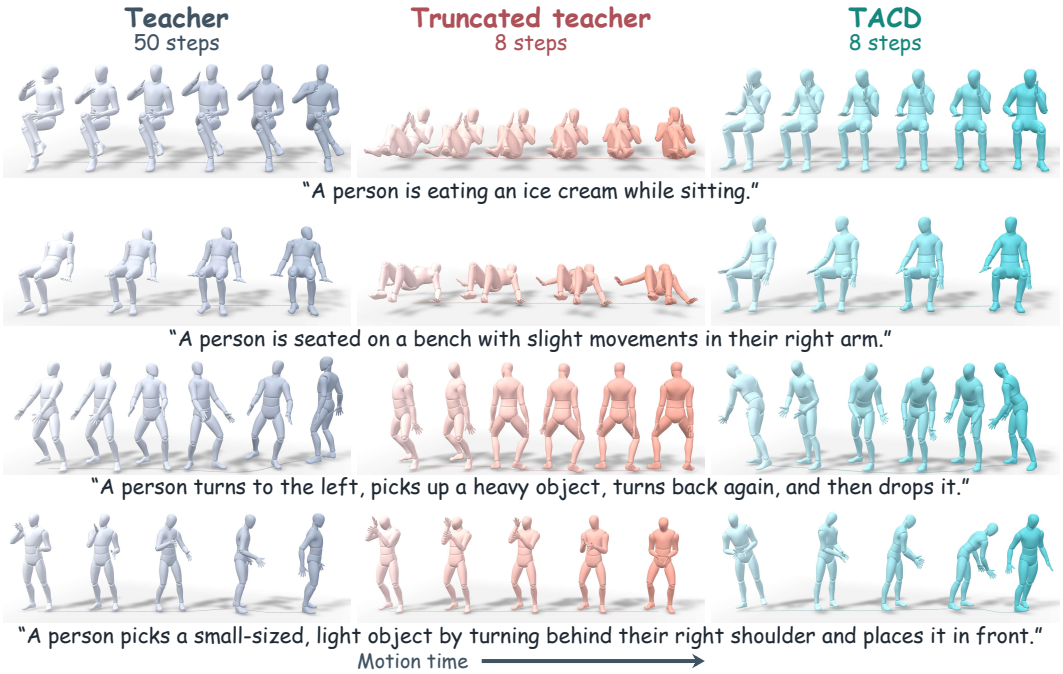


Figure 1: **Motion examples.** Teacher, the same teacher truncated to fewer steps, and TACD. Rows share prompt, noise, frames, and camera (Appendix K.1).

ABSTRACT

Recent text-to-motion models have improved motion quality and instruction following, yet many-step denoising and large model components make deployment slow and memory-intensive. We present Terminal-Amplification-Controlled Distillation (TACD), an on-policy approach for training efficient motion generators from text prompts and pretrained teachers, without real-motion training data. Building on segmented on-policy flow distillation, we supervise clean-motion predictions along student-generated trajectories. We identify a failure mode in which velocity matching on a fixed supervision grid repeatedly overweights errors near the denoising endpoint, degrading few-step generation. TACD ties the latest teacher query to the student's step size, bounding the effective loss weights in clean-motion space without changing inference. Experiments on HumanML3D and KIT-ML demonstrate improved few-step generation, including a 58% reduction in eight-step HY-Motion student FID relative to distillation without this bound. For diffusion teachers, the endpoint-matching form of TACD yields four-step students with lower FID and matched or improved text–motion retrieval relative to their 50-step teachers on HumanML3D. On HY-Motion and Kimodo, eight-step students with compact components achieve 7.7–11.9$\times$ end-to-end speedups and reduce peak GPU memory by 3.8–6.7$\times$ relative to their teachers.

1 INTRODUCTION

Text-to-motion generation provides an intuitive interface for creating human motion from natural-language descriptions, with applications in character animation and interactive virtual environments.

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.

Recent diffusion and flow models have made substantial progress in motion synthesis and instruction following (Tevet et al., 2023; Wen et al., 2025). Yet generation quality alone is insufficient for interactive use: models must also respond promptly to user instructions and operate within limited device memory. These requirements are difficult to meet when generation involves repeated evaluations of a large motion model supported by a heavy text encoder.

Existing acceleration methods replace long denoising procedures with few-step generation. MotionLCM (Dai et al., 2024b) applies latent consistency distillation to motion generation, while MotionPCM (Jiang et al., 2025) introduces phased consistency to improve synthesis quality and efficiency. However, motion-distillation pipelines such as MotionLCM use real motion examples during training, which may be unavailable when adapting a released generator trained on an inaccessible corpus. Moreover, reducing sampling steps alone does not reduce the size of the motion backbone or its conditioning encoder. This motivates learning few-step, compact students from available text prompts and pretrained teachers, without requiring real-motion training data (Gu et al., 2023; Xiang et al., 2025; Zhou et al., 2025).

Teacher-guided on-policy distillation offers a natural starting point by obtaining supervision along trajectories generated by the student. In particular, π-Flow (Chen et al., 2026) learns a policy that can be queried between student network evaluations and matches it to the teacher along its own trajectory. This construction provides teacher feedback on states encountered during generation. Nevertheless, selecting these states does not determine how strongly their prediction errors should be weighted. In our motion-distillation experiments, a student trained with this form of teacher supervision can still exhibit substantial few-step degradation. The central question of this paper is therefore how strongly each supervised state should count.

Our analysis reveals a problematic interaction between the prediction space and the supervision schedule. When clean-motion predictions are trained through velocity matching, their errors receive increasingly large weights near the denoising endpoint. Although this conversion-induced weighting has been studied under random timestep sampling (Hong et al., 2026), a fixed supervision grid queries the same near-terminal time at every training update. The resulting amplification is therefore a persistent component of the loss rather than an occasional heavily weighted sample. Controlled interventions show that bounding this weight restores few-step motion quality and that restoring it reverses the gain (Fig. 2).

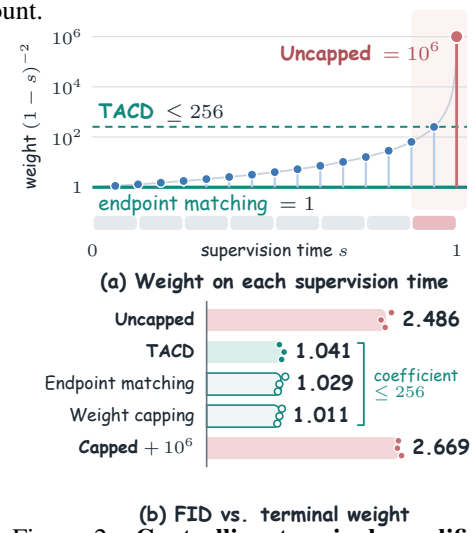


Figure 2: **Controlling terminal amplification improves few-step motion distillation.** **(a)** The known weight $(1-s)^{-2}$ on our supervision grid. **(b)** Bounding it improves FID; hollow bars enforce the same bound on the original grid (dots: seeds).

We introduce *Terminal-Amplification-Controlled Distillation* (TACD), an on-policy distillation method for efficient text-to-motion generation. Building on π-Flow's single-prediction-per-segment construction, the student predicts the entire clean motion at the start of each denoising segment and learns from teacher targets along student-generated trajectories. TACD regulates the resulting supervision weights to prevent excessive terminal amplification: it ties the latest supervision time to the student's step size, which bounds every weight in closed form. Controlled comparisons show that other ways of enforcing the same bound recover comparable quality. The same construction extends to diffusion teachers, for which TACD takes its endpoint-matching form.

On HumanML3D, terminal amplification control reduces the eight-step student's FID from 2.486 to 1.041, and restoring the terminal coefficient alone raises it again to 2.669. Beyond this flow intervention, four-step endpoint-matching students of two diffusion teachers match or improve their teachers on FID and text–motion retrieval on this benchmark, and on KIT-ML and Kimodo (Rempe et al., 2026) the students improve on the teacher truncated to the same step count. Under the SSAE vision–language judge (Wen et al., 2025) on 640 free-form prompts, the eight-step HY-Motion student matches its 50-step teacher and leads seven of ten open-weight HumanML3D models (Table 3); in blind human rankings, its win rate against the teacher exceeds those of the four-step MotionLCM-

V2 and MLCT by 10 and 17 percentage points (Table 4). Efficiency studies on two large generators (Fig. 4) show eight-step students that are **$7.7\times$ (HY-Motion) and $11.9\times$ (Kimodo) faster** end to end than their teachers, with smaller motion backbones and text encoders ($7.0\times$ and $4.7\times$ fewer parameters) reducing **GPU memory by $6.7\times$ and $3.8\times$**.

Our contributions are threefold: (1) we identify terminal amplification in segmented velocity-matching distillation, where a fixed supervision grid places a persistent, orders-of-magnitude larger weight at the same near-terminal supervision time, and show through controlled interventions that it degrades few-step motion quality; (2) we introduce TACD, which ties the latest teacher query to the student's step size and thereby bounds every supervision weight, and extend it to diffusion teachers through its endpoint-matching form; (3) on HumanML3D, KIT-ML, and Kimodo, few-step TACD students improve on the teacher truncated to their step count, and, combined with compact backbones and text encoders, eight-step students run 7.7–$11.9\times$ faster end to end and use 3.8–$6.7\times$ less peak GPU memory than their teachers.

2 Related work

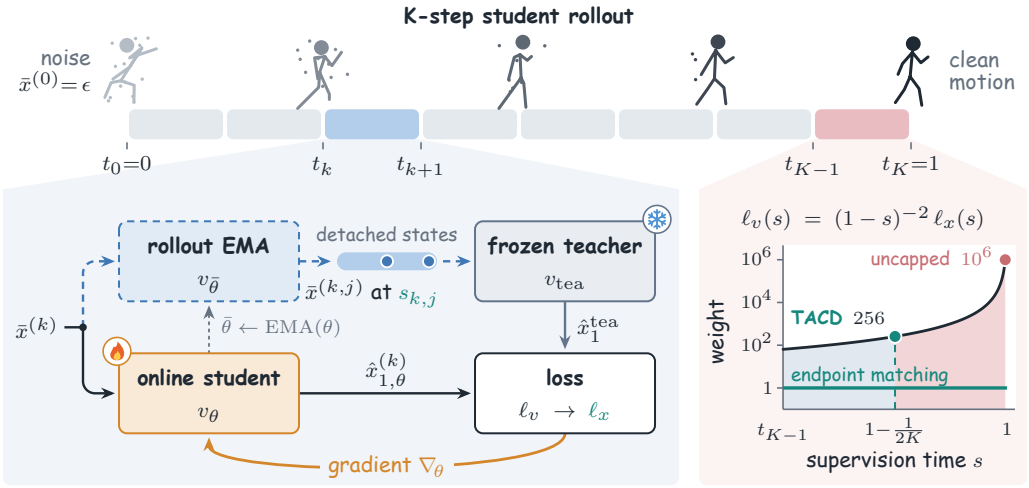
Efficient and data-free motion distillation. Text-to-motion generators include VAEs (Petrovich et al., 2022), token models (Guo et al., 2024; Zhang et al., 2023a), diffusion models (Chen et al., 2023; Tevet et al., 2023), and flow matching (Lipman et al., 2023), as in HY-Motion (Wen et al., 2025). Sampling acceleration includes progressive and consistency distillation (Luo et al., 2023; Salimans & Ho, 2022; Song et al., 2023), distribution and moment matching (Luo et al., 2025; Salimans et al., 2024; Yin et al., 2024a;b), and average-velocity modeling (Geng et al., 2025). MotionLCM (Dai et al., 2024b) uses real motions during distillation; MotionPCM (Jiang et al., 2025) combines phased consistency and adversarial training. We adopt teacher-output supervision without real-motion training targets (Gu et al., 2023; Xiang et al., 2025; Zhou et al., 2025), and evaluate both sampling and model costs; solver and compression methods address complementary inference costs (Amin et al., 2025; Du et al., 2026; He et al., 2024; Li et al., 2023; Lu et al., 2022; Sun et al., 2024).

Policy-based generative distillation. π-Flow (Chen et al., 2026) constructs a queryable DX velocity policy from predicted clean-sample estimates. We use its single-knot specialization: one estimate defines a segment, and teacher queries supervise the student-generated trajectory. Other on-policy distillers study arbitrary-interval flow maps (Gu et al., 2026), flow-matching distillation (Fang et al., 2026b), or black-box adversarial transfer (Luo et al., 2026). Our focus is how the endpoint parameterization interacts with the finite supervision grid of a few-step student.

Prediction–loss alignment and temporal supervision. EDM (Karras et al., 2022) and Min-SNR (Hang et al., 2023) study prediction parameterizations and their induced loss weights. Hong et al. (2026) identify the inverse-square endpoint-to-velocity factor, analyze terminal amplification under random timestep sampling, and study prediction-space losses and capped weights. JiT (Li & He, 2025) clips the conversion denominator; FlowCTS (Ye et al., 2026) identifies concentration from transition-kernel KL weighting. Random time sampling draws a heavily weighted time only occasionally; on a fixed grid over detached student rollouts the same near-terminal supervision time recurs at every update, and its weight becomes a fixed loss coefficient. This permits separate interventions on supervision time and coefficient and a bound tied to the student's step size; TACD's endpoint-matching form for diffusion teachers applies prediction–loss alignment.

3 Method

TACD is a supervision design for segmented on-policy distillation. In each denoising segment the student predicts the complete clean motion once; this prediction fixes the student's states within the segment, and the frozen teacher, queried at those states during training only, supplies the targets that correct it. We first formulate this construction for flow teachers and analyze the effective supervision weights induced by velocity matching. We then introduce terminal amplification control and extend its endpoint-matching formulation to diffusion teachers.



(a) One detached on-policy training step on a segment **(b)** The final segment, K = 8

Figure 3: **Flow training and terminal amplification control.** The rollout EMA generates detached supervision states, while gradients update the online student. TACD moves the supervision times $s_{k,j}$ so that every coefficient is bounded; endpoint matching (ℓ_x) enforces the same bound at the original queries (teal marks what each changes). Both preserve the inference schedule.

3.1 Segmented On-Policy Distillation

Given text c and noise, the student generates a complete motion in K evaluations, without real-motion training targets. Flow time runs from noise at $t = 0$ to clean motion at $t = 1$, and x_t is the entire motion at flow time t, not a physical frame. The *sampling grid* $0 = t_0 < \cdots < t_K = 1$ fixes the K student evaluations and splits denoising into the segments $[t_k, t_{k+1}]$. Following the single-knot DX parameterization of π-Flow (Chen et al., 2026), one clean-motion prediction defines the student policy over each segment: at the segment start $x^{(k)}$, the student's velocity output v_θ gives

$$\hat{x}_{1,\theta}^{(k)} = x^{(k)} + (1 - t_k) v_\theta(x^{(k)}, t_k; c), \tag{1}$$

$$\pi_\theta^{(k)}(x, t) = \frac{\hat{x}_{1,\theta}^{(k)} - x}{1 - t}, \qquad t \in [t_k, t_{k+1}]. \tag{2}$$

The prediction estimates the clean motion at $t = 1$, not the state at the next boundary; holding it fixed within the segment yields the update

$$x^{(k+1)} = x^{(k)} + \frac{t_{k+1} - t_k}{1 - t_k} \left(\hat{x}_{1,\theta}^{(k)} - x^{(k)} \right). \tag{3}$$

Supervision along the student rollout. Teacher supervision is obtained on states that the student itself visits. An EMA copy $\bar{\theta}$ of the student generates the rollout: starting from noise $\bar{x}^{(0)}$, it applies Eqs. (1) and (3) to produce the segment starts $\bar{x}^{(k)}$. Within each segment we place M supervision times

$$s_{k,j}(\delta) = t_k + \frac{j}{M} \left[\min(t_{k+1}, 1 - \delta) - t_k \right], \qquad j = 1, \ldots, M, \tag{4}$$

where the terminal margin $\delta > 0$ keeps the last query below $t = 1$, at which Eq. (2) is undefined. The *supervision grid* $\mathcal{S} = \{s_{k,j}\}$ is used only during training and is distinct from the sampling grid, which always ends at $t_K = 1$. The rollout states at the supervision times are

$$\bar{x}^{(k,j)} = \text{sg}\left[\bar{x}^{(k)} + \frac{s_{k,j} - t_k}{1 - t_k} \left(\hat{x}_{1,\bar{\theta}}^{(k)} - \bar{x}^{(k)} \right) \right], \tag{5}$$

with $\text{sg}[\cdot]$ the stop-gradient. For the loss, the online student computes its prediction $\hat{x}_{1,\theta}^{(k)}$ once at each detached segment start and reuses it for the segment's M queries; the frozen teacher velocity v_{tea} is queried at the supervision states. Velocity matching penalizes, at each query, the residual

$$\ell_v(s_{k,j}) = \left\| \pi_\theta^{(k)}(\bar{x}^{(k,j)}, s_{k,j}) - v_{\text{tea}}(\bar{x}^{(k,j)}, s_{k,j}; c) \right\|_m^2, \qquad \widetilde{\mathcal{L}}_v = \sum_{k=0}^{K-1} \frac{w_k}{M} \sum_{j=1}^{M} \ell_v(s_{k,j}), \tag{6}$$

where w_k are segment weights and $\|\cdot\|_m^2$ is the squared error averaged over valid frames and channels. Rollout states and teacher targets are constants for the update, so gradients pass only through the online predictions at the segment starts. This construction fixes *where* supervision is obtained; the matching space sets how strongly each query counts.

3.2 Terminal amplification in velocity matching

The teacher's instantaneous velocity implies a clean-motion estimate,

$$\hat{x}_1^{\mathrm{tea}}(x,t;c) = x + (1-t)v_{\mathrm{tea}}(x,t;c). \tag{7}$$

Substituting it into the residual of Eq. (6) at a supervision state $\bar{x}$ and time s gives

$$\pi_\theta^{(k)}(\bar{x},s) - v_{\mathrm{tea}}(\bar{x},s;c) = \frac{\hat{x}_{1,\theta}^{(k)} - \hat{x}_1^{\mathrm{tea}}(\bar{x},s;c)}{1-s}, \qquad \text{hence} \qquad \ell_v(s) = (1-s)^{-2}\,\ell_x(s), \tag{8}$$

with $\ell_x(s) = \|\hat{x}_{1,\theta}^{(k)} - \hat{x}_1^{\mathrm{tea}}(\bar{x},s;c)\|_m^2$ the clean-motion loss: velocity matching is clean-motion matching with an inverse-square time-dependent weight (Hong et al., 2026). The objective of Eq. (6) is therefore

$$\mathcal{L}_{\mathcal{S},b} = \sum_{k=0}^{K-1} \frac{w_k}{M} \sum_{j=1}^{M} b_{k,j} \left\| \hat{x}_{1,\theta}^{(k)} - \mathrm{sg}\left[\hat{x}_1^{\mathrm{tea}}(\bar{x}^{(k,j)}, s_{k,j}; c)\right] \right\|_m^2, \qquad b_{k,j} = (1-s_{k,j})^{-2}, \tag{9}$$

where the segment weights w_k are a design choice but the coefficients $b_{k,j}$ are not: they are induced by matching in velocity space.

The last query of the supervision grid sits at $s_{K-1,M} = 1-\delta$ and receives the *terminal coefficient* $b_{K-1,M} = \delta^{-2}$. On a uniform sampling grid every other query satisfies $1 - s_{k,j} \geq 1/(MK)$, so its coefficient is at most $(MK)^2$: a small numerical margin makes one coefficient exceed all others by orders of magnitude ($\delta = 10^{-3}$ gives 10^6). Because the supervision grid is reused at every update, this weight returns at the same supervision time in every update, whatever state the rollout reaches there, instead of being met occasionally as under random time sampling. We call this persistent overweighting *terminal amplification*. A large coefficient does not by itself make its term dominate the update, since each query's gradient share also depends on its residual and on the student Jacobian (Eq. (13)); Section 4.3 measures gradient shares and intervenes on the coefficient directly.

3.3 Terminal amplification control

TACD bounds every coefficient $b_{k,j}$ in Eq. (9) by a value tied to the student's step size. It retains the sampling grid, the inference update, and the training budget, changes only the supervision used during training, and adds no network module or training stage. For a flow teacher it ties the terminal margin in Eq. (4) to the step $\Delta t = 1/K$ of the uniform sampling grid $t_k = k/K$:

$$\delta = \frac{\Delta t}{2} = \frac{1}{2K}, \qquad \text{so that} \qquad \max_{k,j} b_{k,j} = \delta^{-2} = \frac{4}{\Delta t^2} = 4K^2. \tag{10}$$

The latest teacher query, the upper end of the final supervision interval, then sits at the midpoint of the final student step, and Eq. (4) spaces the final segment's M queries uniformly within this shortened interval, so the teacher is never queried next to the singular time, while the sampling grid still ends at $t_K = 1$. The terminal coefficient has the same order as the others on the grid, and the rule is closed-form in the step count, with no tuned threshold. Grid capping is TACD's rule for every flow result: it bounds the weights while keeping the velocity loss of Eq. (9) unchanged.

Two other enforcements meet the same bound at the original queries, each with a different loss; they serve as controls in Section 4.3, and one of them, endpoint matching, is the form TACD takes for diffusion teachers (Section 3.4). Weight capping truncates the coefficient, $b_{k,j} = \min\{(1-s_{k,j})^{-2}, 4K^2\}$, the truncated-weight form of Hong et al. (2026) related to denominator clipping (Li & He, 2025), which with $M = 2$ changes a single term of Eq. (9); endpoint matching sets $b_{k,j} \equiv 1$, which matches the teacher's clean-motion estimates directly and applies prediction–loss alignment (Hong et al., 2026) to fixed-grid on-policy supervision. The three enforcements differ in how they meet the bound: grid capping moves the final-segment queries, weight capping lowers one coefficient, and endpoint matching removes the conversion factor at every query.

3.4 Extension to diffusion teachers

For a diffusion teacher the student again predicts clean motion at each segment start and learns from frozen-teacher targets on detached rollout states, with two changes: the states follow deterministic DDIM sub-trajectories, and the loss is Eq. (9) with $b_{k,j} \equiv 1$. Let $\tau_0 > \cdots > \tau_K$ be the teacher sampler's K-step timestep schedule with cumulative signal level $\bar{\alpha}_\tau$, and $\hat{x}_{0,\theta}^{(k)}$ the student's clean-motion prediction at the segment start $\bar{x}^{(k)}$ (x_0 is the flow case's x_1 under reversed time). Holding the rollout prediction $\hat{x}_{0,\bar{\theta}}^{(k)}$ fixed over the segment also fixes the noise $\bar{\epsilon}^{(k)}$ it implies at $\bar{x}^{(k)}$, and the DDIM states at M supervision times $\tau_{k,j}$ within the segment,

$$\bar{x}^{(k,j)} = \mathrm{sg}\left[\sqrt{\bar{\alpha}_{\tau_{k,j}}}\,\hat{x}_{0,\bar{\theta}}^{(k)} + \sqrt{1-\bar{\alpha}_{\tau_{k,j}}}\,\bar{\epsilon}^{(k)}\right], \tag{11}$$

take the place of Eq. (5). The target is the frozen teacher's clean-motion estimate $\hat{x}_0^{\mathrm{tea}}$ (its clean-sample output, or the DDIM predicted sample of a noise-prediction teacher), computed with the teacher sampler's guidance (Eq. (17) in Appendix A). Written in clean-motion space from the start, this loss carries no $(1-s)^{-2}$ factor; matching noise predictions on the same states would instead weight the clean-motion error by the signal-to-noise ratio (Hang et al., 2023; Salimans & Ho, 2022), which also grows without bound toward the clean end (Eq. (18)). Endpoint matching thus carries no time-dependent weight. Grid capping acts on the flow conversion factor and is specific to flow teachers; for diffusion teachers TACD denotes this endpoint-matching form. Teacher-specific segment weights and rollout settings are reported with each experiment; pseudocode and implementation settings are in Appendices A and C.

4 Experiments

On HumanML3D, the HY-Motion flow teacher tests whether TACD improves few-step students (Section 4.2) and whether the terminal coefficient is the cause (Section 4.3). Section 4.4 tests the endpoint-matching form with the MLD and MotionLCM-V2 diffusion teachers and adds KIT-ML (MDM teacher) and Kimodo (compact encoder); Section 4.5 measures deployment cost.

4.1 Experimental setup

HumanML3D and KIT-ML use dataset-specific Guo evaluators and real-motion references. The main flow student, TACD (HY-Motion), pairs the 0.46B motion backbone of the released HY-Motion Lite with frozen Qwen3-0.6B conditioning and a trainable projection. Controls share initialization, 6k prompts, 8k updates, eight-step inference, CFG 5, and final evaluation-EMA checkpoints; only supervision changes. They use $M = 2$ supervision times per segment, segment weights $w_k = 1$, and, on the original supervision grid, the terminal margin $\delta = 10^{-3}$. The diffusion students are teacher-initialized, low-rank adapted, and trained with DDIM endpoint matching (Section 3.4) at two, four, and eight steps. Training uses training-split captions as text only, with teacher supervision and no real-motion targets; evaluation uses the test split. Table 2 repeats the HY-Motion and MLD comparisons with the MARDM evaluator (Meng et al., 2025b) (protocol in Appendix E.3).

Kimodo (Rempe et al., 2026) students keep its 0.28B backbone with LoRA and use its 8B encoder or a trained-in 1B encoder, TACD (Kimodo), evaluated on 607 captions with Kimodo's TMR evaluator. Full settings and evaluation details are in Appendices C and F.

Metrics. Following Guo et al. (2022), FID compares generated and real motions in a pretrained feature space, R-precision (R@k) and MM-Dist measure text–motion alignment, and Diversity (Div; $\rightarrow$: closer to real is better) and MultiModality (MMod) measure variation across and within prompts; Kimodo uses TMR-based retrieval and FID (Petrovich et al., 2023). NFE counts guided denoising steps per motion (sampling steps; a guided step counts once); definitions are in Appendix C.2.

4.2 Few-step motion quality

On HumanML3D, at the same eight-step Lite budget, TACD reduces mean FID from 2.486 to 1.041 and improves R@3 from .708 to .782 (Table 1); the uncapped student uses π-Flow's single-knot DX

	Method	NFE	R@1↑	R@3↑	MM-Dist↓	FID↓	Div→	MMod↑
Real-motion training	*Published generators (real-motion training) and real-motion reference*							
	Real motions (Guo et al., 2022)	–	.511	.797	2.974	.002	9.503	–
	MDM (fixed eval.) (Tevet et al., 2023)	∼1000	.418	.707	3.63	.489	9.45	2.87
	T2M-GPT (Zhang et al., 2023a)	–	.492	.775	3.121	.141	9.722	1.831
	MoMask (full) (Guo et al., 2024)	–	.521	.807	2.958	.045	–	1.241
	IRG-MotionLLM S3 (Li et al., 2026b)	–	.535	.820	2.785	.242	9.900	–
	MotionHiFlow (Li et al., 2026a)	–	.563	.843	2.691	.032	9.504	–
	MoGeFlow-L (Fang et al., 2026a)	–	.592	.873	2.599	.058	9.981	–
	*Published few-step generators (real-motion training; * released checkpoint, our evaluation)*							
	MotionLCM-V2 student* (Dai et al., 2024a)	4	.554	.841	2.760	.072	9.54	1.72
	MLCT (Hu et al., 2024)	4	.537	.826	2.819	.060	9.545	2.571
	MotionPCM (Jiang et al., 2025)	4	.559	.842	2.716	.030	9.575	1.714
HY-Motion teacher	*Our evaluation: HY-Motion references*							
	HY-Motion teacher	50	.455	.743	3.471	.541	9.13	3.28
	HY-Motion Lite (released)	50	.435	.728	3.575	.726	8.95	–
	Truncated teacher	8	.378	.652	4.225	3.250	8.24	–
	Controlled Lite students, three-seed means: Qwen3-0.6B encoder, teacher-only supervision							
	Uncapped student (DX, $N=1$)	8	.402	.708	3.703	2.486	8.01	**3.994**
	TACD	8	**.479**	**.782**	**3.192**	**1.041**	8.92	3.541
	Non-adversarial ports to the Lite student, three-seed means: teacher-only supervision							
	DMD port	8	.426	.730	3.493	1.829	8.22	3.023
	Consistency port	8	.394	.699	3.754	2.908	7.77	3.803
	Reflow port	8	.370	.660	4.063	3.341	8.07	3.886

Table 1: **HumanML3D.** At eight steps with the HY-Motion teacher, TACD is best in R@1, R@3, MM-Dist, and FID.

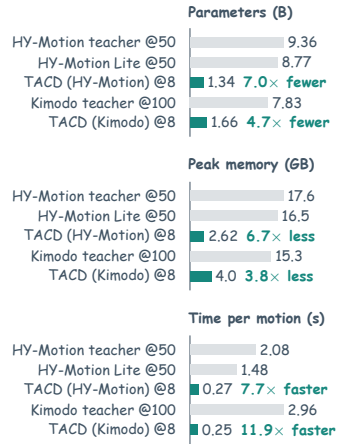


Figure 4: **Deployment cost**, RTX L40, batch 1 (Appendix J); ×: vs. the teacher; @N: N steps.

Method	NFE	R@1↑	R@2↑	R@3↑	MM-Dist↓	FID↓	Div→	CLIP↑
Real motions	–	.505	.702	.801	3.241	–	10.33	.638
HY-Motion flow teacher; students at eight steps, three-seed means								
HY-Motion teacher	50	.415	.601	.708	4.056	3.119	9.47	.604
HY-Motion Lite (released)	50	.394	.578	.685	4.243	3.741	9.29	.589
Truncated teacher	8	.369	.544	.646	4.721	6.816	8.92	.510
Uncapped student (DX, $N=1$)	8	.343	.528	.647	4.562	7.559	8.24	.569
TACD	8	.412	.606	.721	3.955	3.575	9.21	.621
DMD port	8	.371	.559	.678	4.299	5.789	8.67	.591
Consistency port	8	.325	.505	.624	4.697	7.317	8.11	.563
Reflow port	8	.321	.492	.604	4.980	10.473	8.21	.526
MLD diffusion teacher; TACD at four steps, three-seed mean								
MLD teacher (DDIM)	50	.483	.680	.780	3.339	.512	10.35	.630
Truncated MLD teacher	4	.447	.645	.754	3.556	.735	10.31	.599
TACD	4	.507	.703	.801	3.215	.228	10.70	.639

Table 2: **MARDM evaluator** (Meng et al., 2025b): TACD leads the truncated teacher and every eight-step student in R@k, MM-Dist, FID, and CLIP score; four-step MLD TACD matches real motions in R-precision.

Model	SSAE↑	Δ vs. TACD [95% CI]
HY-Motion teacher @50	.695	−.025 [−.053, +.004]
HY-Motion Lite (released) @50	.692	−.028 [−.057, +.001]
Truncated teacher @8	.587	−.133 [−.166, −.100]
Uncapped student @8	.650	−.070 [−.093, −.045]
DMD port @8	.672	−.048 [−.071, −.024]
Consistency port @8	.700	−.020 [−.044, +.006]
Reflow port @8	.590	−.130 [−.156, −.102]
TACD @8	**.720**	–
Open-weight HumanML3D models, all rendered from joints		
MDM @1000	.588	−.112 [−.141, −.080]
MLD @50	.641	−.058 [−.089, −.026]
T2M-GPT	.618	−.081 [−.112, −.051]
MoMask	.636	−.063 [−.093, −.033]
MARDM	.642	−.057 [−.086, −.028]
MotionHiFlow	.679	−.021 [−.052, +.010]
MoGeFlow-L	.716	+.017 [−.013, +.046]
MotionLCM @4	.661	−.038 [−.068, −.007]
MLCT @4	.666	−.033 [−.062, −.004]
MotionLCM-V2 @4	.697	−.002 [−.031, +.029]
TACD @8, from joints	.699	–

Table 3: **SSAE:** a VLM judge's yes-rate on questions about the prompted motion (Appendix G); lower block: Δ vs. TACD from joints.

Configuration	Final-segment queries	C	FID	seed range	vs. uncapped
Uncapped student	0.937, 0.999	10^6	2.486	[2.419–2.563]	–
❶ **TACD** (grid capping)	0.90625, 0.9375	256	1.041	[1.009–1.074]	−58%
❷ Endpoint matching	0.937, 0.999	1	1.029	[0.988–1.091]	−59%
❸ Weight capping	0.937, 0.999	256	1.011	[0.980–1.044]	−59%
Last-point-only	0.937, 0.9375	256	1.039	[1.032–1.048]	−58%
Capped grid, C restored to 10^6	0.90625, 0.9375	10^6	2.669	[2.645–2.703]	+7%

Table 5: **Bounding the terminal coefficient** C, eight-step Lite: each bound (❶–❸) cuts FID by 58–59%; restoring $C = 10^6$ (rose) undoes it. Brackets: three-seed range.

Model	Above teacher (%)↑ [95% CI]	Prompts W/T/L
HY-Motion teacher @50	50 (reference)	–
MotionLCM-V2 @4	41.2 [30.4, 51.7]	17/0/33
MLCT @4	33.9 [24.5, 43.4]	11/0/39
TACD @8	**50.8** [41.1, 60.5]	28/1/21

Table 4: **Human preference:** win rate against the 50-step teacher and prompts won/tied/lost (Appendix H).

policy on a fixed supervision grid (Appendix B), so the comparison evaluates terminal amplification control. The ports, our reimplementations of DMD (Yin et al., 2024b), consistency distillation (Luo et al., 2023), and Reflow (Liu et al., 2023) on the same Lite student, share its architecture, prompts, and generator-update budget (Appendix D); Reflow gives FID 3.341 and R@3 .660, no better than the teacher truncated to eight steps (3.250/.652). Under the MARDM evaluator (Table 2), TACD leads the truncated teacher and every eight-step student in each seed and cuts the uncapped student's FID from 7.559 to 3.575.

TACD (HY-Motion) exceeds the R@3 of the 50-step teacher (.743) and of the released Lite at 50 steps (.728), while the teacher retains better FID and diversity. On SSAE, a VLM-judged semantic-alignment score (Table 3), TACD scores highest (.720): .020–.028 above the 50-step teacher, the released Lite, and the Consistency port and .05–.13 above the DMD, truncated-teacher, uncapped, and Reflow students. When every model, TACD included (.699), is rendered from joints on one skeleton, TACD leads seven of ten open-weight HumanML3D models by .03–.11, including the

Method	Training signal	NFE	R@1↑	R@3↑	MM-Dist↓	FID↓	Div→	MMod↑
MLD teacher (Chen et al., 2023) *and MotionLCM-V1* (Dai et al., 2024b)								
MLD teacher (DDIM)	Reference	50	.502	.795	3.048	.387	9.53	–
Truncated MLD teacher	None	4	.459	.768	3.295	.585	9.49	–
MotionLCM-V1	Real motions	4	–	.798	–	.286	–	–
TACD	Teacher outputs	4	**.525**	**.818**	**2.909**	**.222**	9.80	–
MotionLCM-V2 teacher and student (Dai et al., 2024a)								
MotionLCM-V2 teacher	Reference	50	.537	.824	2.838	.066	9.59	–
MotionLCM-V2 student	Real motions	4	**.554**	**.841**	**2.760**	.072	9.54	1.72
TACD	Teacher outputs	4	.538	.824	2.838	**.055**	9.57	**1.94**

Table 6: **Diffusion teachers on HumanML3D.** Four-step TACD has lower FID than its 50-step teacher and the released MotionLCM student (our evaluation); bold: best at four steps in each group.

four-step MotionLCM and MLCT, and shows no detectable difference from MoGeFlow-L, MotionHiFlow, and MotionLCM-V2. In a blind ranking study (Table 4), annotators rank eight-step TACD above its 50-step teacher in 51% of comparisons, on par with it, while the four-step MotionLCM-V2 and MLCT reach 41% and 34%. For the teacher, the best FID and R@3 over CFG $\{2, 3, 5, 7.5\}$ both occur at 5, the value TACD (HY-Motion) uses (Table 8).

4.3 Supervision analysis and controlled interventions

At $K = 8$ and $M = 2$, the original supervision grid queries the final segment at .937 and .999: the terminal coefficient is 10^6, while the other fifteen coefficients are at most 252. We first measure where the gradient goes, then intervene on this one coefficient (Tables 5 and 17).

The final segment takes most of the gradient norm. We measure $\rho_k = \|\partial\mathcal{L}_k/\partial\theta\| / \sum_r \|\partial\mathcal{L}_r/\partial\theta\|$ on the full Lite backbone before clipping. At initialization, the original supervision grid assigns 68.3% of the summed segment-gradient norms to the final segment, versus $.88\%$ on the capped grid; the uncapped 8k checkpoint reaches 96.6%. Probing both checkpoints on both grids (Table 17), the original grid puts 96.6–97.6% of the gradient norm on the final segment and the capped grid 3.6–11.8%, so concentration follows the grid rather than the trained weights.

Bounding the coefficient restores quality; restoring it reverses the gain. All three enforcements of the bound of Section 3.3 recover FID to 1.01–1.04 (Table 5; per-seed values in Table 16): grid capping (❶, TACD), endpoint matching (❷, the form used for diffusion teachers), and weight capping (❸), which sets the terminal coefficient to $C = 4K^2 = 256$ on the original supervision grid and leaves every other term unchanged (Eq. (15)). Conversely, restoring $C = 10^6$ on the capped grid, without moving any query, worsens FID in all three seeds, to a mean of 2.669.

One query carries the effect. Keeping .937 and moving only .999 to .9375 gives FID 1.039, close to full capping's 1.041, with improvement in all three seeds (with the 1B student at four steps: 3.291 uncapped, 1.458 fully capped, 1.462 with only the last point moved, Appendix I).

Relative weight, not total loss scale. The coefficient also changes the magnitude of the loss, but the optimizer removes that difference. Gradient clipping at unit norm binds on at least 799 of the 800 logged updates in every run of Table 5: the median pre-clip norm is 245–263 for the uncapped student and 5.0–6.1 for the bounded students (Table 17). Clipping normalizes the gradient magnitude entering AdamW, so the interventions act through the relative supervision weights (the terminal share of Table 17).

Sensitivity to the terminal coefficient. Sweeping the terminal coefficient (Fig. 6; every run in Appendix I.4) gives a dose–response: FID stays at 1.01–1.07 from $C = 256$ to 4096 and rises monotonically beyond it, to 1.14, 1.30, 1.70, and 2.49 at $C = 16384$, 65536, 262144, and 10^6, while the median pre-clip norm rises in step from 6 to 253. The other metrics move with FID at every step, and moving the grid instead ($\delta \in \{1/32, 3/32\}$) stays in the plateau: $C = 4K^2$ sits in a plateau at least $16\times$ wide.

Key finding. In fixed-grid flow distillation, velocity matching can repeatedly overweight errors near the denoising endpoint. TACD controls this amplification with a step-size-aware bound, reducing eight-step HY-Motion student FID by **58% versus uncapped distillation** on HumanML3D without changing inference. Restoring the full weight reverses the gain.

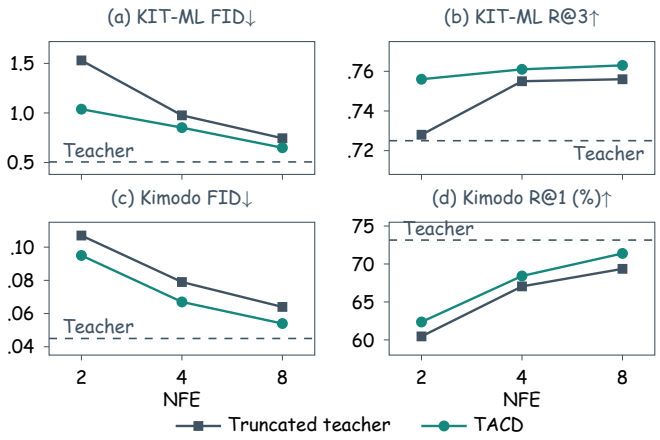


Figure 5: **Few-step curves.** On KIT-ML (MDM teacher) and Kimodo, TACD (teal) beats the truncated teacher (grey) at every NFE; dashed: the full-step teacher.

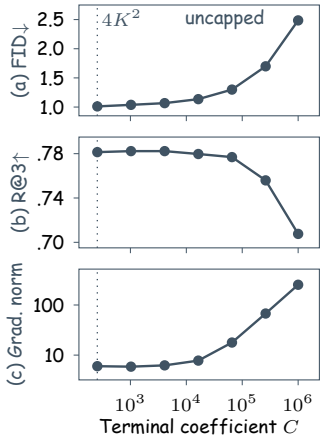


Figure 6: **Sweeping** C on the eight-step Lite student: TACD's $4K^2$ (dotted) sits on the plateau.

4.4 Other teachers and datasets

Diffusion teachers on HumanML3D. With endpoint matching, four-step TACD improves on the 50-step MLD teacher in both FID (.387 $\rightarrow$ .222) and R@3 (.795 $\rightarrow$ .818), and its FID is below the .286 we measure for the released MotionLCM-V1 checkpoint, a student trained on real motions (Table 6). With the MotionLCM-V2 teacher, four-step TACD reaches FID .055 against the teacher's .066 and the released official student's .072 under the same evaluation, at the teacher's R@3 (.824). TACD FID is the mean of three training seeds ($\pm$.002 for MLD, $\pm$.001 for MotionLCM-V2). Under the MARDM evaluator, four-step TACD has FID .228 against the MLD teacher's .512 and matches real motions in R-precision (Table 2).

KIT-ML. At the same NFE, TACD reduces FID from the truncated teacher's 1.530, .976, and .745 to 1.038, .852, and .649 at two, four, and eight steps (Fig. 5a; Table 10b), 32% lower at two steps and 13% at four and eight; our MDM teacher run agrees with its published row (Table 10a).

Kimodo. At eight steps, TACD raises R@1 over the truncated teacher from 69.36% to 71.39% and lowers FID from .064 to .054, with the same ordering at four and two steps (Fig. 5c,d; Table 13).

4.5 Quality–efficiency trade-offs

Table 13 and Fig. 4 separate the savings from fewer sampling steps (computation) and from smaller backbones or encoders (memory). On HY-Motion, the Lite backbone and Qwen3-0.6B encoder of TACD (HY-Motion) reduce peak GPU memory from 17.63 to 2.62 GB (6.7×), mostly from the 8B-to-0.6B text encoder, and eight-step sampling cuts end-to-end time on an RTX L40 (`bf16`, batch 1) from the teacher's 2077 to 268 ms (7.7×); the released Lite at 50 steps takes 1476 ms (Table 19).

TACD (Kimodo), with a trained-in 1B encoder, reduces peak memory from 15.3 to 4.0 GB (3.8×) and end-to-end time from 2.96 to .25 s (11.9×) at the R@3 and FID of the 8B-encoder student (Tables 13 and 14). Figure 1 shows sample motions (Appendix K.1).

5 Conclusion

TACD trains efficient text-to-motion students from text prompts and pretrained teachers alone. Where supervision is taken does not settle how strongly each error counts: on a fixed grid, velocity matching overweights the near-terminal query at every update, and a step-size bound on that weight restores few-step quality; compact encoders and backbones then cut latency and memory.

AI USE STATEMENT

AI tools were used to polish manuscript wording and LaTeX formatting, assist with literature retrieval and discovery, and generate synthetic data used in our experiments.

REPRODUCIBILITY STATEMENT

The supplementary material provides complete flow pseudocode and objective definitions (Appendix A), training and evaluation settings (Appendix C), Kimodo evaluation and compact-encoder comparisons (Appendix F), and per-seed controls and gradient probes (Appendix I).

REFERENCES

Farhana Amin, Sabiha Afroz, Kanchon Gharami, Mona Moghadampanah, and Dimitrios S. Nikolopoulos. DiffPro: Joint timestep and layer-wise precision optimization for efficient diffusion inference. *arXiv preprint arXiv:2511.11446*, 2025.

Hansheng Chen, Kai Zhang, Hao Tan, Leonidas Guibas, Gordon Wetzstein, and Sai Bi. pi-flow: Policy-based few-step generation via imitation distillation. In *ICLR*, 2026. arXiv:2510.14974.

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 *CVPR*, 2023.

Wenxun Dai, Ling-Hao Chen, Yufei Huo, Jingbo Wang, Jinpeng Liu, Bo Dai, and Yansong Tang. Real-time controllable motion generation via latent consistency model. MotionLCM-V2 technical report, 2024a. https://dai-wenxun.github.io/files/MotionLCM_V2.pdf.

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 *ECCV*, 2024b.

Jinyang Du, Shenghao Jin, Ziqian Xu, Ruihao Gong, Shiqiao Gu, Yang Yong, Jinyang Guo, and Xianglong Liu. Collaborative few-step distillation and low-bit quantization for Wan2.2 dual-expert video diffusion models. *arXiv preprint arXiv:2606.00658*, 2026.

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

Zhen Fang, Wenxuan Huang, Yu Zeng, Yiming Zhao, Shuang Chen, Kaituo Feng, Yunlong Lin, Lin Chen, Zehui Chen, Shaosheng Cao, and Feng Zhao. Flow-OPD: On-policy distillation for flow matching models. *arXiv preprint arXiv:2605.08063*, 2026b.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv:2505.13447*, 2025.

Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Lingjie Liu, and Joshua M. Susskind. Boot: Data-free distillation of denoising diffusion models with bootstrapping, 2023. arXiv:2306.05544.

Yuchao Gu, Guian Fang, Yuxin Jiang, Weijia Mao, Song Han, Han Cai, and Mike Zheng Shou. AnyFlow: Any-step video diffusion model with on-policy flow map distillation. *arXiv preprint arXiv:2605.13724*, 2026.

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 *CVPR*, 2022.

Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. In *CVPR*, 2024.

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *ICCV*, 2023.

Yefei He, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. EfficientDM: Efficient quantization-aware fine-tuning of low-bit diffusion models. In *International Conference on Learning Representations (ICLR)*, 2024.

Jiadong Hong, Lei Liu, Xinyu Bian, Wenjie Wang, and Zhaoyang Zhang. Prediction–loss alignment for sampler–robust flow matching training. *arXiv preprint arXiv:2602.10420*, 2026.

Mengxian Hu, Minghao Zhu, Xun Zhou, Qingqing Yan, Shu Li, Chengju Liu, and Qijun Chen. Efficient text-driven motion generation via latent consistency training. *arXiv preprint arXiv:2405.02791*, 2024.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. In *NeurIPS*, 2023.

Lei Jiang, Ye Wei, and Hao Ni. Motionpcm: Real-time motion synthesis with phased consistency model. *arXiv preprint arXiv:2501.19083*, 2025.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022.

Heng Li, Xiaotong Lin, Ling-An Zeng, Yulei Kang, Shuai Li, and Jian-Fang Hu. MotionHiFlow: Text-to-motion via hierarchical flow matching. In *CVPR*, 2026a. arXiv:2604.23264.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025.

Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-Diffusion: Quantizing diffusion models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Yuan-Ming Li, Qize Yang, Nan Lei, Shenghao Fu, Ling-An Zeng, Jian-Fang Hu, Xihan Wei, and Wei-Shi Zheng. IRG-MotionLLM: Interleaving motion generation, assessment and refinement for text-to-motion generation. In *ECCV*, 2026b. arXiv:2512.10730.

Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. In *WACV*, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *ICLR*, 2024.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv:2310.04378*, 2023.

Yang Luo, Shengju Qian, Xiaohang Tang, Zirui Zhu, Yong Liu, Xin Wang, and Yang You. On-policy adversarial flow distillation for autoregressive video generation. *arXiv preprint arXiv:2605.26105*, 2026.

Yihong Luo, Tianyang Hu, Jiacheng Sun, Yujun Cai, and Jing Tang. Learning few-step diffusion models by trajectory distribution matching. *arXiv:2503.06674*, 2025.

Zichong Meng, Zeyu Han, Xiaogang Peng, Yiming Xie, and Huaizu Jiang. Absolute coordinates make motion generation easy. *arXiv preprint arXiv:2505.19377*, 2025a.

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 *CVPR*, 2025b.

Mathis Petrovich, Michael J. Black, and Gül Varol. Temos: Generating diverse human motions from textual descriptions. In *ECCV*, 2022.

Mathis Petrovich, Michael J. Black, and Gül Varol. TMR: Text-to-motion retrieval using contrastive 3D human motion synthesis. In *ICCV*, 2023.

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

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

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022.

Tim Salimans, Thomas Mensink, Jonathan Heek, and Emiel Hoogeboom. Multistep distillation of diffusion models via moment matching. *arXiv:2406.04103*, 2024.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, 2023.

Haojun Sun, Chen Tang, Zhi Wang, Yuan Meng, Jingyan Jiang, Xinzhu Ma, and Wenwu Zhu. TMPQ-DM: Joint timestep reduction and quantization precision selection for efficient diffusion models. *arXiv preprint arXiv:2404.09532*, 2024.

Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H. Bermano. Human motion diffusion model. In *ICLR*, 2023.

Yuxin Wen, Qing Shuai, Di Kang, Jing Li, Cheng Wen, Yue Qian, Ningxin Jiao, Changhai Chen, Weijie Chen, Yiran Wang, Jinkun Guo, Dongyue An, Han Liu, Yanyu Tong, Chao Zhang, Qing Guo, Juan Chen, Qiao Zhang, Youyi Zhang, Zihao Yao, Cheng Zhang, Hong Duan, Xiaoping Wu, Qi Chen, Fei Cheng, Liang Dong, Peng He, Hao Zhang, Jiaxin Lin, Chao Zhang, Zhongyi Fan, Yifan Li, Zhichao Hu, Yuhong Liu, Linus, Jie Jiang, Xiaolong Li, and Linchao Bao. Hy-motion 1.0: Scaling flow matching models for text-to-motion generation. *arXiv:2512.23464*, 2025.

Qianlong Xiang, Miao Zhang, Yuzhang Shang, Jianlong Wu, Yan Yan, and Liqiang Nie. DKDM: Data-free knowledge distillation for diffusion models with any architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Kaiyang Ye, Yuan Ge, Junxiang Zhang, Bei Li, Ziming Zhu, Haishu Zhao, Xiaoqian Liu, Chenglong Wang, Jingbo Zhu, Zhengtao Yu, and Tong Xiao. Flowcts: On-policy continuous trajectory supervision of flow models. *arXiv preprint arXiv:2607.24522*, 2026.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *NeurIPS*, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, 2024b.

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Yong Zhang, Hongwei Zhao, Hongtao Lu, Xi Shen, and Ying Shan. Generating human motion from textual descriptions with discrete representations. In *CVPR*, 2023a.

Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Remodiffuse: Retrieval-augmented motion diffusion model. In *ICCV*, 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 TPAMI*, 2024.

Mingyuan Zhou, Zhendong Wang, Huangjie Zheng, and Hai Huang. Guided score identity distillation for data-free one-step text-to-image generation. In *International Conference on Learning Representations (ICLR)*, 2025.

SUPPLEMENTARY MATERIAL

METHOD

A Objective and implementation details **15**
A.1 Objective definitions . . . 15
A.2 Training algorithm and conventions . . . 15
A.3 Diffusion-teacher branch . . . 16

B Relation to the single-knot π-Flow objective **17**

EXPERIMENTAL SETUP

C Experimental settings **17**
C.1 Models and training . . . 17
C.2 Evaluation . . . 18
C.3 Training configurations . . . 19

D Baseline implementations **19**

RESULTS

E Results by benchmark **20**
E.1 HumanML3D with the flow teacher . . . 20
E.2 Diffusion teachers on HumanML3D . . . 20
E.3 A second HumanML3D evaluator . . . 21
E.4 KIT-ML . . . 21

F Kimodo **21**
F.1 Step compression and conditioning . . . 22
F.2 Training on the 1B encoder . . . 22
F.3 Held-out split analysis . . . 22

G Semantic fidelity under a VLM judge **22**

H Human ranking study **23**

I Controlled interventions and gradient statistics **23**
I.1 Supervision-grid control on the 1B student . . . 24
I.2 Coefficient interventions by seed . . . 24
I.3 Gradient probes and clipping . . . 24
I.4 Terminal-coefficient sweep . . . 24

J Deployment cost **25**
J.1 Latency and memory . . . 25

K Qualitative results **28**
K.1 Teacher, truncated teacher, and TACD . . . 28
K.2 Eight-step gallery . . . 28

A OBJECTIVE AND IMPLEMENTATION DETAILS

This appendix expands Section 3 into what the trainer computes: the objectives that Section 3 uses without displaying (Appendix A.1), the flow training iteration and its conventions (Appendix A.2), and the diffusion-teacher branch (Appendix A.3).

A.1 OBJECTIVE DEFINITIONS

The objects below enter training in this order: the rollout prediction, the gradient form of the conversion factor, and the two loss variants used in the experiments.

Rollout prediction. The EMA student $\bar{\theta}$ generates the rollout with the same one-step parameterization as the online student (Eq. (1)). At each rollout segment start $\bar{x}^{(k)}$ it predicts

$$\hat{x}^{(k)}_{1,\bar{\theta}} = \bar{x}^{(k)} + (1 - t_k) v_{\bar{\theta}}(\bar{x}^{(k)}, t_k; c), \tag{12}$$

without gradient. This prediction fixes the next segment start through Eq. (3) and the supervision states through Eq. (5).

Gradient form of the conversion. The supervision state and the teacher target are both detached, so the factor $(1-s)^{-2}$ of Eq. (8) does not depend on θ and the conversion holds for gradients, query by query:

$$\nabla_\theta \ell_v(t) = \frac{1}{(1-t)^2} \nabla_\theta \ell_x(t). \tag{13}$$

A query's gradient is its clean-motion gradient scaled by $(1-t)^{-2}$. How much the scaled query contributes to an update therefore also depends on the size of $\nabla_\theta \ell_x$, which Section 4.3 measures.

Endpoint matching. Setting every coefficient $b_{k,j}$ of Eq. (9) to one gives

$$\widetilde{\mathcal{L}}_x(\theta; \bar{\theta}) = \sum_{k=0}^{K-1} \frac{w_k}{M} \sum_{j=1}^{M} \left\| \hat{x}^{(k)}_{1,\theta} - \mathrm{sg}\left[\hat{x}^{\mathrm{tea}}_1(\bar{x}^{(k,j)}, s_{k,j}; c) \right] \right\|^2_m, \tag{14}$$

which compares the student's clean-motion prediction directly with the teacher's clean-motion estimates at the supervision states. It is the third enforcement of the bound in Section 3.3 and the form TACD takes for diffusion teachers (Section 3.4).

Coefficient interventions. The interventions of Section 4.3 and Appendix I.4 change the weight of a single query. A weight $a_{k,j}$ on the velocity loss of query (k, j) acts as the endpoint coefficient

$$a_{k,j} \ell_v(s_{k,j}) = \underbrace{\frac{a_{k,j}}{(1-s_{k,j})^2}}_{\text{endpoint-equivalent coefficient}} \ell_x(s_{k,j}), \tag{15}$$

so $a_{k,j} = C(1-s_{k,j})^2$ gives that query the coefficient C while every other term and the normalization w_k/M stay unchanged. Weight capping is the case $C = 4K^2$ on the terminal query.

A.2 TRAINING ALGORITHM AND CONVENTIONS

Algorithm 1 gives one training iteration of the flow branch: TACD's grid capping, or the endpoint-matching control of Section 4.3. The paragraphs after it state the conventions the iteration relies on: fixed supervision times, the endpoint form of the residual, and the rollout EMA.

Supervision times are a fixed grid, not samples. The loss of Eq. (6) and the diffusion objective in Eq. (17) are computed with $M{=}2$ deterministic supervision points per segment, $s_{k,j} = t_k + (s^{\mathrm{hi}}_k - t_k)\frac{j}{M}$ for $j{=}1, 2$, so no randomness enters beyond the noise ϵ and the prompt. For the flow case $s^{\mathrm{hi}}_k = \min(t_{k+1}, 1-\delta)$: the policy velocity $\pi_\theta(x_t, t) = (\hat{x}^{(k)}_1 - x_t)/(1-t)$ is singular as $t \to 1$, and the last segment's upper endpoint is excluded by the margin δ. The uncapped student uses $\delta = 10^{-3}$; TACD uses $\delta = 1/(2K)$ on a uniform sampling grid (Table 7). The last-point-only control moves only the last supervision time and leaves the other original point unchanged.

The flow residual is the endpoint residual, reweighted. Equation (8) holds pointwise and for the same norm, so Eq. (6) is an endpoint-matching loss carrying an implicit $(1-t)^{-2}$ weight. The flow endpoint-matching control removes this factor on the same states (Section 4.3); Eq. (17) applies endpoint matching on the DDIM trajectory. The norm is a mask-averaged squared error over valid frames and channels, with a factor $\frac{1}{2}$ in the flow branch.

Rollout EMA. The rollout that fixes the segment starts $\bar{x}^{(k)}$ and the supervision states $\bar{x}^{(k,j)}$ comes from the EMA weights $\bar{\theta}$ and is detached; the frozen teacher supplies the targets, and gradients pass only through the online student's segment-start endpoints (Eq. (6)). The diffusion objective in Eq. (17) uses the same detached-state convention on its DDIM trajectory. The rollout-EMA decay is μ=0.999 on the flow teacher and μ=0.98 on Kimodo; the evaluation EMA is 0.999 everywhere.

A.3 Diffusion-teacher branch

This subsection expands Section 3.4. For the ϵ- and x_0-prediction diffusion teachers, each segment is a deterministic (η=0) DDIM sub-trajectory, with M supervision timesteps $\tau_{k,j} = \tau_{k+1} + \frac{j}{M}(\tau_k - \tau_{k+1})$, $j = 1, \dots, M$, and $\tau_K = 0$; the last is the segment start. The student commits to one $\hat{x}_0^{(k)}$ per segment from a single gradient-carrying evaluation at the segment start, and the supervision states lie on the *detached* DDIM sub-trajectory induced by that constant $\hat{x}_0^{(k)}$, built with the noise implied by the rollout prediction at the segment start,

$$\bar{\epsilon}^{(k)} = \frac{\bar{x}^{(k)} - \sqrt{\bar{\alpha}_{\tau_k}}\,\hat{x}_{0,\bar{\theta}}^{(k)}}{\sqrt{1-\bar{\alpha}_{\tau_k}}}. \tag{16}$$

The student matches the teacher's per-step $\hat{x}_0$ at those states,

$$\mathcal{L}_{\text{on}}^{x_0} = \sum_k \frac{w_k}{M} \sum_j \left\| \hat{x}_{0,\theta}^{(k)} - \text{sg}[\hat{x}_0^{\text{tea}}(\bar{x}^{(k,j)}, \tau_{k,j}; c)] \right\|_m^2, \tag{17}$$

which is Eq. (9) with $b_{k,j} \equiv 1$; gradients pass only through the segment-start predictions. Matching noise predictions on the same states would instead weight the clean-motion error by the signal-to-

Algorithm 1: One training iteration of the flow branch. TACD uses $q = \text{grid}$ (capped supervision grid, velocity loss); $q = \text{endpoint}$ is the endpoint-matching control (original supervision grid, endpoint loss). Only the online student receives gradients.

Input: frozen teacher v_{tea}; online student θ; rollout EMA $\bar{\theta}$; boundaries $\{t_k\}_{k=0}^K$; control $q \in \{\text{grid}, \text{endpoint}\}$

Choose $\delta \leftarrow 1/(2K)$ if $q = \text{grid}$, else $\delta \leftarrow 10^{-3}$; construct $\{s_{k,j}(\delta)\}$ with Eq. (4)

sample noise $\bar{x}^{(0)}$ and text c

`// detached rollout-EMA trajectory`

for $k = 0, \dots, K-1$ **do**

  compute $\hat{x}_{1,\bar{\theta}}^{(k)}$ with Eq. (12)

  store $\bar{x}^{(k,j)}$ with Eq. (5) for $j = 1, \dots, M$

  apply the update in Eq. (3) with $\hat{x}_{1,\bar{\theta}}^{(k)}$ to obtain $\bar{x}^{(k+1)}$

end

$\mathcal{L} \leftarrow 0$

`// online prediction and frozen-teacher queries`

for $k = 0, \dots, K-1$ **do**

  compute $\hat{x}_{1,\theta}^{(k)}$ at detached $\bar{x}^{(k)}$

  for $j = 1, \dots, M$ **do**

    query $\hat{x}_1^{\text{tea}}(\bar{x}^{(k,j)}, s_{k,j}; c)$ with Eq. (7)

    set $\ell_q^{(k,j)} \leftarrow \ell_x(s_{k,j})$ if $q = \text{endpoint}$, else $\ell_v(s_{k,j})$, using the pointwise residuals in Eq. (8)

    $\mathcal{L} \leftarrow \mathcal{L} + \frac{w_k}{M}\ell_q^{(k,j)}$

  end

end

update θ; update rollout EMA $\bar{\theta}$ and the separate evaluation EMA

noise ratio,

$$\left\|\hat{\epsilon}_\theta - \hat{\epsilon}^{\text{tea}}\right\|_m^2 = \frac{\bar{\alpha}_\tau}{1-\bar{\alpha}_\tau}\left\|\hat{x}_{0,\theta}^{(k)} - \hat{x}_0^{\text{tea}}\right\|_m^2, \tag{18}$$

which grows without bound toward the clean end; Eq. (17) carries no such weight.

Time direction. Flow time runs $t_0{=}0 \to t_K{=}1$ from noise to data, so segment k is $[t_k, t_{k+1}]$ with $t_k{<}t_{k+1}$. DDIM timesteps run the other way, $\tau_0{>}\tau_1{>}\ldots$, so the same segment is written $[\tau_{k+1}, \tau_k]$ in Eqs. (11) and (17). Both denote the same object within the k-th sampler step.

Schedule and guidance in the diffusion branch. $\bar{\alpha}_\tau$ at a continuous τ is a linear interpolation of the teacher's own `alphas_cumprod` between $\lfloor\tau\rfloor$ and $\lceil\tau\rceil$; teacher and student share that array, the same K-step timestep schedule, and the same classifier-free guidance scale applied identically to both (a guided evaluation counts once toward NFE, Appendix C.2).

B Relation to the single-knot π-Flow objective

The flow objective of Section 3.1 uses the $N{=}1$ single-knot DX policy of π-Flow (Chen et al., 2026), written in our time convention, on a fixed supervision grid rather than π-Flow's uniformly sampled query times.

Single-knot window identity. Along the rollout induced by one fixed endpoint, the policy velocity $(\hat{x}_1 - x_t)/(1-t)$ is constant, so π-Flow's micro-window averaging (the policy velocity averaged over $[t, t{+}\Delta t]$) reduces to the instantaneous velocity.

Numerical check. On random `float64` states, the trainer's segment loss and its gradient with respect to the student's endpoint agree with an independent implementation of the $N{=}1$ DX segment loss to 10^{-9} relative tolerance, and the window average computed by quadrature matches the instantaneous value to the same tolerance. The uncapped student in Table 1 therefore uses π-Flow's single-knot DX policy on the fixed supervision grid.

C Experimental settings

This appendix collects the settings shared by the experiments of Section 4: models and training for each benchmark (Appendix C.1), the evaluation protocol and model selection (Appendix C.2), and the named training configurations (Appendix C.3).

C.1 Models and training

We evaluate few-step distillation on HumanML3D (Guo et al., 2022) and KIT-ML (Plappert et al., 2016), using HY-Motion flow and MDM diffusion teachers, respectively. Kimodo provides a second core system evaluation of few-step generation and compact conditioning under a TMR-based protocol. The controlled HumanML3D comparison uses HY-Motion's frozen 1B flow teacher with its native Qwen3-8B text encoder. The student combines the 0.46B Lite motion backbone with a frozen Qwen3-0.6B encoder and a trainable conditioning projection. The uncapped student, TACD, and the two other enforcements of its bound share initialization, a 6,000-prompt cache, 8,000 training iterations, learning rate 3×10^{-6}, effective batch size 4, and evaluation EMA selection at the final iteration. All use uniform eight-step inference and CFG 5.0. They differ only in the supervision grid, matching space, or terminal coefficient defined in Sections 3.3 and 4.3. Configurations and port-specific settings are in Table 7 and Appendix D.

On KIT-ML, the teacher is the released 1000-step MDM (Tevet et al., 2023), which predicts x_0 in a 251-d raw-motion representation. We train teacher-initialized, rank-32 LoRA students at two, four, and eight steps with the DDIM endpoint-matching recipe in Section 3.4, using trailing timestep spacing (Lin et al., 2024) for the student and the truncated teacher alike. Training uses 4,000 text prompts, batch size 32, learning rate 10^{-4}, CFG 2.5, rollout EMA, and 10,000 iterations. Our students on both benchmarks train with teacher supervision and no real motions.

For Kimodo (Rempe et al., 2026), the teacher is a released 100-step x_0-diffusion model over a 77-joint skeleton. The student retains the teacher's 0.28B motion backbone and uses low-rank adaptation. TACD (Kimodo) conditions on a Llama-3.2-1B encoder through a layer-fusion head and motion adapters. We use DDIM endpoint matching with the teacher's 8B text encoder at $K \in \{8, 4, 2\}$. Evaluation uses 607 prompts in a TMR-based harness. Of these, 567 have ground-truth motions, used for evaluation only; stratum results and training settings are in Appendix F. Generation latency is measured end-to-end for eight-second motions on one RTX L40 (Appendix J). All 8B-encoder rows have 15.3 GB measured peak memory. In the trained-in encoder comparison, both encoders train with global batch 64 at $K = 8$ (the 8B encoder split across 2–4 GPUs by data parallelism, drawing the same global minibatch). Complete paired records are in Table 14.

C.2 Evaluation

HumanML3D and KIT-ML use the dataset-specific evaluators of Guo et al. (2022): a text encoder and a motion encoder trained so that matching text–motion pairs lie close in a shared feature space. The quality, alignment, and diversity metrics below are computed on these features, with the real test motions as the reference, and averaged over 20 evaluator repetitions; HumanML3D uses $\text{rerank}_k = 1$ and `fp32` evaluation. Appendix E.3 repeats the HumanML3D comparisons with the 67-dimensional evaluators of Meng et al. (2025b). In the tables, $\uparrow$ and $\downarrow$ mark the better direction, and $\rightarrow$ means closer to real motions is better.

Motion quality. FID is the Fréchet distance between Gaussians fitted to the features of the generated motions and of the real test motions.

Text alignment. Evaluation texts and their generated motions are split into groups of 32. R-precision R@k is the fraction of texts whose own motion is among the k motions of the group closest to the text in feature space. MM-Dist is the mean Euclidean distance between a text's feature and the feature of the motion generated for it. The CLIP score of Table 2 is the cosine similarity between a generated motion and its caption under the CLIP-based evaluator of Meng et al. (2025b).

Diversity. With v the feature of a generated motion, $\text{Div} = \frac{1}{S_d}\sum_{i=1}^{S_d} \|v_i - v'_i\|_2$ over $S_d = 300$ random pairs of generated motions, compared with the value of real motions (9.503 on HumanML3D). $\text{MMod} = \frac{1}{CI}\sum_{c=1}^{C}\sum_{i=1}^{I} \|v_{c,i} - v'_{c,i}\|_2$ measures variation within a prompt: $C = 100$ prompts, 30 motions each, and $I = 10$ random pairs per prompt.

Kimodo. Texts and motions are embedded with Kimodo's released TMR model (Petrovich et al., 2023), trained for its skeleton. For each of the 607 prompts (BONES-SEED captions), R@k ranks all 607 generated motions by similarity to the prompt; a motion generated for a near-identical caption (TMR text-embedding cosine above 0.96) also counts as correct. FID is the Fréchet distance between the embeddings of the generated motions and of the captions in TMR's joint space, so it uses no real motions and is not comparable with the HumanML3D FID. GT-R@1 is R@1 restricted to the 567 prompts that have a ground-truth motion.

Cost. NFE counts guided denoising steps per motion, i.e., sampling steps: a classifier-free-guided step counts once for every method, whether its conditional and unconditional branches run in one batched call (HY-Motion, the latent-diffusion teachers MLD and MotionLCM-V2, and Kimodo) or in two (MDM, for teacher and student alike). Time and memory are measured end to end, guidance included (Appendix J). The VLM-judged SSAE score is defined in Appendix G and the human ranking in Appendix H.

Seeds and intervals. Primary HumanML3D student rows average three training seeds (42/43/44). On KIT-ML, TACD point estimates also average three training seeds; the $\pm$ values beside evaluator metrics are 95% intervals over the 20 evaluator repetitions, not training-seed variation, and per-seed FID is in Table 11 (eight-step standard deviation 0.008). Gradient probes instead repeat minibatch sampling on fixed weights.

Model selection. Every trained row of Tables 1 and 12, ours and ported alike, is the *final iterate of a fixed iteration budget* (the artifact at `max_iters`), not a validation-FID-selected checkpoint.

C.3 TRAINING CONFIGURATIONS

Table 7 names every training configuration used in the paper.

Purpose	Student / encoder	K	Loss; sampling grid; margin δ	Replication
Flow controls	1B LoRA-32 / 8B	4	Uncapped/grid cap/ last-point; uniform; .001/.125	3 each
Lite controls	Lite full / 0.6B	8	Original/grid cap/ endpoint/last-point; uniform; .001/.0625	3 each
Coefficient interventions	Lite full / 0.6B	8	Weight cap on original grid; C=10^6 on capped grid	3 each
Gradient probe	Lite full / 0.6B	8	Uncapped/grid cap; uniform; .001/.0625	32 batches $\times$ 2 probe seeds
DDIM transfer	Teacher-size / native	4	Endpoint matching; native DDIM grid	MLD/V2: 3

Table 7: **Experimental configurations.** HY-Motion 1B is the teacher for the first four rows; replication counts denote training seeds except for gradient probes; text encoders remain frozen. Teal rows contain the TACD recipes: grid capping for the flow teacher, endpoint matching for diffusion teachers.

The terminal-amplification analysis is evaluated on the flow branch, where the endpoint-to-velocity conversion induces the time-dependent coefficient; the diffusion experiments use the endpoint-matching formulation of Section 3.4. The flow interventions are evaluated on uniform sampling grids at 1B/$K = 4$ and Lite/$K = 8$, with three training seeds per configuration, and the bound is the closed-form midpoint rule $\delta = 1/(2K)$, $C = 4K^2$.

D BASELINE IMPLEMENTATIONS

We detail the ported baselines of Table 1. Each port keeps its method-specific targets, auxiliary networks, and hyperparameters, and shares the student architecture, prompt cache, seeds, generator-update budget, and evaluation protocol.

The three port rows use the same student as the TACD rows: the 0.46B Lite motion head with the Qwen3-0.6B conditioning path and its initialization, the same 6,000-prompt cache, K=8, CFG 5.0 at sampling, AdamW with generator learning rate 3×10^{-6}, batch 2 with 2-step gradient accumulation, 8,000 generator updates, seeds 42/43/44, and the final checkpoint. Objective-specific settings are listed below.

DMD port. *Objective:* distribution matching (Yin et al., 2024a;b) ported to the flow teacher. The generator's backward-simulated clean motion is trained against the difference between the frozen teacher score and a learned fake score, with the fake network updated five times per generator update at learning rate 10^{-5}, together with the regression term of DMD on the teacher's own (noise, 50-step output) pairs with weight 1. No real motions enter either term.

Consistency port. HY-Motion's training motions are not released, so the port first synthesizes teacher motions and then applies consistency distillation (Luo et al., 2023) to that corpus. It uses the teacher's 50-step CFG-5 outputs on the same prompt cache with fresh noise, 32 consistency boundaries inside the eight-step grid, one CFG-5 teacher Euler substep per target, a Consistency target EMA (decay .95), and a pseudo-Huber loss ($c = .01$). CFG is folded into the student during training, so the student samples at CFG 1, as in LCM.

Reflow port. *Objective:* rectified-flow distillation (Liu et al., 2023; 2024) on the teacher's own couplings. For each of the same 6,000 prompts, a noise sample ϵ and the teacher's 50-step CFG-5 output y_T define $x_t = (1 - t)\epsilon + t\, y_T$, and the student's CFG-combined velocity regresses $y_T - \epsilon$ at one stratified t per segment of the eight-step grid. No real motions enter. Every other setting is inherited from the TACD rows unchanged (initialization, prompt cache, learning rate, batch, 8,000 updates, EMA, seeds 42/43/44, final checkpoint), so only the method-specific training objective and target construction differ from TACD.

E RESULTS BY BENCHMARK

This appendix gives the complete tables behind the quality comparisons of Sections 4.2 and 4.4, one subsection per benchmark: HumanML3D with the flow teacher (Appendix E.1), HumanML3D with diffusion teachers (Appendix E.2), HumanML3D under a second evaluator (Appendix E.3), and KIT-ML (Appendix E.4); Kimodo follows in Appendix F.

E.1 HUMANML3D WITH THE FLOW TEACHER

Beyond Table 1, this subsection gives published HumanML3D context and the teacher's guidance sweep (Table 8).

Published HumanML3D context. Table 1 retains the reported results of generators trained on real motions. MLCT and MotionPCM are source-reported; the MotionLCM-V1 and V2 rows throughout this paper are the released checkpoints generated by us and evaluated in our harness.

Model	NFE	CFG	FID↓	R@3↑	MM-Dist↓
Teacher (1B), 50 steps	50	2.0	0.9937	0.7052	3.7308
Teacher (1B), 50 steps	50	3.0	0.6989	0.7312	3.5619
Teacher (1B), 50 steps	50	5.0	0.5407	0.7431	3.4706
Teacher (1B), 50 steps	50	7.5	0.5487	0.7408	3.4893
Teacher (1B), 8 steps (truncated)	8	5.0	3.2503	0.6516	4.2250
Student (0.46B), uncapped (s42)	8	5.0	2.5631	0.7056	3.7131
Student (0.46B), uncapped (s43)	8	5.0	2.4189	0.7082	3.6962
Student (0.46B), uncapped (s44)	8	5.0	2.4755	0.7089	3.6995
mean of the three seeds	8	5.0	2.4859	0.7076	3.7029
TACD, 0.46B (s42)	8	5.0	1.0085	0.7829	3.1811
TACD, 0.46B (s43)	8	5.0	1.0737	0.7813	3.1999
TACD, 0.46B (s44)	8	5.0	1.0399	0.7821	3.1955
mean of the three seeds	8	5.0	1.0407	0.7821	3.1922
Endpoint matching, 0.46B (3-seed mean)	8	5.0	1.0292	0.7812	3.1876
Real motions (reference)	–	–	–	0.8075	2.9453

Table 8: **Retrieval–quality trade-off: eight-step Lite vs. the 50-step HY-Motion teacher.** Student rows: Qwen3-0.6B encoder, full 0.46B Lite backbone; teacher rows: its 8B encoder. Mean rows average three seeds; CFG 5.0 is the teacher's best FID and R@3 of four tested. Rose rows: terminal coefficient 10^6.

Teacher guidance sweep. TACD's three-seed R@3 is 0.7821 (range 0.7813–0.7829), MM-Dist is 3.1922 (3.1811–3.1999), FID is 1.0407 (1.0085–1.0737), and diversity is 8.9186 (8.8859–8.9444). The corresponding teacher readings are 0.7431, 3.4706, 0.5407, and 9.1315. The teacher's best observed FID and R@3 both use CFG 5.0 over $\{2.0, 3.0, 5.0, 7.5\}$; Table 8 preserves the complete sweep.

E.2 DIFFUSION TEACHERS ON HUMANML3D

Table 6 puts both diffusion teachers next to their released students on one metric protocol, and Table 9 gives the full MLD and MotionLCM-V2-teacher comparisons. At matched NFE (K=4) TACD's FID is below the V2 student's (0.055 vs. 0.072) with higher MultiModality.

(a) Original MLD teacher (the teacher MotionLCM-V1 distills)

Method	NFE	FID↓	R@3↑	Div→
Teacher (MLD, 50)	50	0.387	0.795	9.53
Truncated teacher	4	0.585	0.768	9.49
MotionLCM-V1 (consistency)	4	0.286	0.798	–
TACD	4	0.222 ±.002	0.818	9.80

(b) Retrained V2 teacher (the teacher of the V2 student)

Method	NFE	FID↓	R@3↑	Div→	MMod↑
MotionLCM-V2 (consistency)	4	0.072	0.841	9.54	1.72
TACD	4	0.055 ±.001	0.824	9.57	1.94

Table 9: **MotionLCM comparison**: *(a)* the MLD teacher, *(b)* the MotionLCM-V2 teacher. MotionLCM rows: released checkpoints, our harness. TACD rows: three-seed mean±std.

E.3 A SECOND HUMANML3D EVALUATOR

Table 2 rescores the generated motions of Tables 1 and 6 with the released 67-dimensional evaluators and evaluation code of Meng et al. (2025b) over 20 repetitions, keeping each test item's caption and length from our Guo protocol; on real test motions the loader reproduces the reference values of Meng et al. (2025a) (R@3 .795; MM-Dist 3.241 vs. 3.244). Under this evaluator TACD (HY-Motion) remains the best eight-step system on every metric in each of its three seeds.

E.4 KIT-ML

Method	NFE	R@1↑	R@3↑	MM-Dist↓	FID↓	Div→
(a) Published generators (real-motion training)						
Real motions (Guo et al., 2022)	–	.424	.779	2.788	.031	11.08
T2M (Guo et al., 2022)	–	.361	.681	3.488	3.022	10.72
MotionDiffuse (Zhang et al., 2024)	–	.417	.739	2.958	1.954	11.10
MDM† (Tevet et al., 2023)	∼1000	.404	.731	3.096	.513	10.73
MLD (Chen et al., 2023)	50	.390	.734	3.204	.404	10.80
T2M-GPT (Zhang et al., 2023a)	–	.416	.745	3.007	.514	10.92
ReMoDiffuse (Zhang et al., 2023b)	–	.427	.765	2.814	.155	10.80
MotionGPT (Jiang et al., 2023)	–	.366	.680	3.527	.510	10.35
MoMask (Guo et al., 2024)	–	.433	.781	2.779	.204	–
MotionPCM (Jiang et al., 2025)	4	.443	.789	2.881	.336	10.76
IRG-MotionLLM S3 (Li et al., 2026b)	–	.445	.781	2.740	.432	11.12

Method	NFE	R@1↑	R@3↑	MM-Dist↓	FID↓	Div→
(b) Our evaluation: MDM teacher, teacher-only students						
Real motions (GT)	–	.426	.780	2.786	.027	10.99
MDM teacher	1000	.403	.725	3.115	.506	10.74
Truncated teacher	2	.395	.728	3.193	1.530	11.24
TACD	2	.410	.756	2.996	1.038	11.06
Truncated teacher	4	.410	.755	3.014	.976	11.38
TACD	4	.416	.761	2.932	.852	11.16
Truncated teacher	8	.416	.756	2.948	.745	11.34
TACD	8	.421	.763	2.873	.649	11.08

Table 10: **KIT-ML, MDM teacher.** TACD beats the truncated teacher at every NFE in R@1, R@3, MM-Dist, and FID; TACD rows: three-seed means after 10k updates; intervals in Table 11. †Post-fix evaluation endorsed by the MDM authors (repository issue #211).

Table 10b evaluates the released MDM teacher (Tevet et al., 2023), an x_0-prediction, 251-d raw-motion diffusion model, with TACD's DDIM endpoint-matching recipe at two, four, and eight steps. The measured block uses one evaluation harness: the 1000-step teacher has FID 0.506 ± 0.023 and R@3 0.725 ± 0.005. Few-step samplers use trailing timestep spacing (Lin et al., 2024), which places the first step at $t = 999$. At the same NFE, endpoint matching reduces FID by 32% at two steps and 13% at four and eight.

Evaluation conventions. Published rows (Table 10a) retain their sources' conventions. Our evaluation takes the caption length before padding, the corrected path of the released MDM code; the re-measured teacher row is within about one reported confidence interval of the corrected MDM reference on all four checked metrics. Table 11 repeats the measured rows with their 95% intervals over the 20 evaluator repetitions and the per-seed FID of the TACD rows. GT is recomputed in each run; the displayed GT row comes from the teacher run.

Method	NFE	R@1↑	R@3↑	MM-Dist↓	FID↓	Div→	FID by seed
Real motions (GT)	–	.426	.780 ±.005	2.786	.027 ±.004	10.99	–
MDM teacher	1000	.403	.725 ±.005	3.115	.506 ±.023	10.74	–
Truncated teacher	2	.395	.728 ±.005	3.193	1.530 ±.063	11.24	–
TACD	2	.410	.756	2.996	1.038	11.06	1.020 / 1.077 / 1.017
Truncated teacher	4	.410	.755 ±.006	3.014	.976 ±.064	11.38	–
TACD	4	.416	.761	2.932	.852	11.16	.840 / .857 / .859
Truncated teacher	8	.416	.756 ±.006	2.948	.745 ±.051	11.34	–
TACD	8	.421	.763	2.873	.649	11.08	.649 / .656 / .641

Table 11: **KIT-ML measured rows with evaluator intervals** (the rows of Table 10; Guo evaluator, 20 repetitions). ±: 95% interval over the 20 repetitions. TACD rows average seeds $42/43/44$; the last column lists their per-seed FID.

F KIMODO

The main-text Kimodo result is in Section 4.4 and Table 13; this appendix gives the full track, the encoder comparison, and a held-out split analysis. Kimodo is a 100-step x_0-diffusion teacher over a 77-joint skeleton, trained on BONES-SEED; TACD uses none of its motions. All comparisons use a 607-prompt BONES-SEED caption set scored with Kimodo's released TMR evaluator, and GT-R@1 is R@1 restricted to the 567 prompts that have a ground-truth motion.

F.1 Step compression and conditioning

Table 12 gives the full Kimodo track: TACD with the teacher's 8B encoder at $K \in \{8, 4, 2\}$, and the teacher's own DDIM at the same budgets; Table 13 adds TACD (Kimodo).

Method	NFE	R@1 (%)↑	R@3 (%)↑	FID↓	GT-R@1 (%)↑	Lat.(s)↓	VRAM
Teacher	100	73.15	88.63	0.045	75.66	2.96	15.3G
Truncated teacher	8	69.36	84.35	0.064	72.66	0.29	15.3G
Truncated teacher	4	67.05	82.37	0.079	70.55	0.18	15.3G
Truncated teacher	2	60.46	77.27	0.107	63.14	0.13	15.3G
TACD, 8B enc.	8	71.39 ±.50	86.55 ±.25	0.054	75.01 ±.37	0.29	15.3G
TACD, 8B enc.	4	68.42 ±.69	83.25 ±.38	0.067	72.13 ±.77	0.18	15.3G
TACD, 8B enc.	2	62.36 ±.12	79.57 ±.23	0.095	65.43 ±.00	0.13	15.3G

Table 12: **Kimodo results** (607-caption TMR evaluation; GT-R@1: R@1 restricted to the 567 prompts that have a ground-truth motion). ±: SD over three seeds (K=2: two). TACD rows: teacher's 8B encoder. Lat./VRAM: end to end, one RTX L40.

Step compression transfers: TACD at K=8 reaches R@1 71.39±0.50 / R@3 86.55±0.25 / FID 0.0542 against the teacher's 73.15 / 88.63 / 0.045. The teacher's own DDIM degrades gracefully—69.36 / 67.05 / 60.46 at K=8 / 4 / 2—and TACD leads it by +2.03 / +1.37 / +1.89 R@1, and each of its eight training runs (three seeds at $K = 8/4$, two at $K = 2$) exceeds the truncated teacher at the same step count in R@1, R@3, and FID.

The conditioning axis transfers too: a student conditioned on Llama-3.2-1B through a distilled layer-fusion head (no motion data anywhere) cuts VRAM 15.3→4.0 GB. The two efficiency axes separate: of the end-to-end 11.9× against the 100-step teacher (2.96 s→0.25 s), step compression accounts for 10.1× (2.96 s→292 ms with the 8B encoder at the student's own step count) and the encoder change for 1.18× (292→248 ms); at K=8 the denoiser, not the text encoder, is most of the wall clock. VRAM is entirely the encoder (3.8×), since step count does not move a sampler's peak. All timings are medians over five round-robin passes on one RTX L40 (Appendix J). Table 14 trains the student on the 1B encoder outright, against the 8B encoder under matched seeds.

F.2 Training on the 1B encoder

The TACD rows of Table 12 were trained with the teacher's own 8B encoder. Table 14 trains the student end-to-end on the encoder it runs with: three seeds per cell at K=8, with the same code, schedule, and generation settings.

Swapping the 8B encoder for the trained-in 1B encoder leaves FID unchanged (+0.0002, against per-seed standard deviations of at most 0.0003) and R@3 flat. The 8B-encoder rows of Table 14 are separate training runs from those of Table 12.

F.3 Held-out split analysis

Of the 567 GT-bearing prompts, 462 caption BONES-SEED clips in the teacher's training split and 105 are held out; these clips are used only for evaluation, never for student training. Restricted to each stratum, TACD @8 and the truncated teacher@8 score 74.0/71.9 (teacher-train) and 88.6/84.8 (held-out) GT-R@1: on the held-out prompts TACD keeps its advantage (+3.8 pp, against +2.2 pp on the rest).

G Semantic fidelity under a VLM judge

This appendix details the VLM-judged comparison of Section 4.2, which scores semantic fidelity through questions about the described motion rather than through a learned embedding. It follows the SSAE construction of HY-Motion (Wen et al., 2025), with our own prompt set and judge models. Table 3 scores eight HY-Motion systems and ten open-weight HumanML3D models with a vision–language judge on a fixed set of 640 prompts in six categories: 600 captions of BONES-SEED motion clips, tagged generic, compositional, common, long, or rare-verb, and 40 prompts we wrote outside that corpus. None comes from HumanML3D, whose texts the HY-Motion students train on. The released Lite row uses its own Qwen3-8B encoder at 50 steps; the ports are the students of Table 1, the Consistency port sampled at its own CFG of 1.

Method	Encoder	NFE	R@1 (%)↑	R@3 (%)↑	FID↓	Time (s)↓	Memory (GB)↓
Teacher	8B	100	73.15	88.63	0.045	2.96	15.3
Truncated teacher	8B	8	69.36	84.35	0.064	0.29	15.3
TACD	8B	8	71.39	86.55	0.054	0.29	15.3
TACD (Kimodo)	1B	8	68.86	86.66	0.054	0.25	4.0
Truncated teacher	8B	4	67.05	82.37	0.079	0.18	15.3
TACD	8B	4	68.42	83.25	0.067	0.18	15.3
TACD (Kimodo)	1B	4	67.00	82.98	0.066	0.13	4.0
Truncated teacher	8B	2	60.46	77.27	0.107	0.13	15.3
TACD	8B	2	62.36	79.57	0.095	0.13	15.3

Table 13: **Kimodo teacher** (607-caption TMR evaluation). TACD with the 8B encoder beats the truncated teacher at every NFE; at NFE 8, TACD (Kimodo) with a 1B text encoder keeps R@3 and FID with 3.8× less memory (paired runs in Table 14).

Conditioning encoder	NFE	R@1 (%)↑	R@3 (%)↑	FID↓	GT-R@1 (%)↑
teacher's own 8B encoder	8	71.72 ±.50	86.66 ±.33	0.0542 ±.0002	75.25 ±.10
1B encoder, trained in	8	68.86 ±.44	86.66 ±.28	0.0544 ±.0003	72.60 ±.67
paired Δ (1B − 8B)	8	−2.86 [−3.30, −2.47]	−0.00 [−0.66, +0.49]	+0.0002 [−0.0001, +0.0004]	−2.65 [−3.18, −1.94]

Table 14: **Training on the 1B text encoder vs. the teacher's own 8B encoder** (Kimodo, 607-prompt bench, K=8; the two conditions share seeds 42/43/44). ±: SD over three seeds; brackets on the Δ rows: range of the three paired differences.

Open-weight models. Each model uses its released HumanML3D checkpoint with its own default sampler and guidance, one fixed seed, and six-second clips: MDM (Tevet et al., 2023) at 1000 steps, the MLD teacher of Table 6 (Chen et al., 2023) with 50 DDIM steps, T2M-GPT (Zhang et al., 2023a), MoMask (Guo et al., 2024), MARDM (Meng et al., 2025b), MotionHiFlow (Li et al., 2026a), MoGeFlow-L (Fang et al., 2026a) with 96 flow steps, and the four-step MotionLCM (Dai et al., 2024b), MLCT (Hu et al., 2024), and MotionLCM-V2 (Dai et al., 2024a). Their HumanML3D features are decoded to joint positions, retargeted onto the HY-Motion skeleton by matching bone directions (mean error below 4°), and rendered with the same renderer, camera, and resolution as the HY-Motion rows; the last two frames of MARDM outputs, which contain a decoding jump, are held at the preceding frame. For this block TACD goes through the same path (.699; .720 when rendered directly), so the lower block of Table 3 compares against TACD rendered from joints.

Protocol. For each prompt, gpt-5.5 authored binary sub-questions about the described motion ("does the person raise both arms?"), and gpt-5.4-mini gated them by answerability from video. Each generation was rendered as a six-second clip, of which the judge (gpt-5.6-luna) saw 14 frames spread over the clip (384 px); it answered every question three times, a question counting when all three votes are usable and its answer being the majority vote. A system's score is its yes-rate over the 2,571 gated questions (636 prompts) that all eight HY-Motion systems cover; differences are paired by prompt with a 4,000-sample prompt-cluster bootstrap.

H Human Ranking Study

The study behind Table 4 compares four systems on 50 prompts drawn at random from the 640 of Appendix G, stratified by prompt category: eight-step TACD, the 50-step HY-Motion teacher, and the four-step open-weight models MotionLCM-V2 and MLCT, all rendered from joints as in the lower block of Table 3. Each page shows the prompt and the four six-second clips playing together; the systems are hidden and their screen positions are balanced within each annotator. Annotators rank the clips primarily by how well the motion follows the prompt, using motion naturalness only to break ties. Table 4 reports, for each system, the share of rankings that place it above the teacher, and the prompts on which most annotators place it above (won), below (lost), or equally often (tied). 17 annotators contributed 691 rankings; confidence intervals come from 4,000 bootstrap samples that resample annotators and prompts together.

I Controlled Interventions and Gradient Statistics

This appendix gives the per-seed records and gradient statistics behind Section 4.3: the supervision-grid control on the 1B student (Appendix I.1), the coefficient interventions by seed (Appendix I.2), the gradient probes and the role of clipping (Appendix I.3), and the full terminal-coefficient sweep (Appendix I.4).

I.1 SUPERVISION-GRID CONTROL ON THE 1B STUDENT

Table 15 repeats the supervision-grid control of Section 4.3 on the 1B student at $K{=}4$: the seed-matched uncapped and capped pair (three seeds each, identical except for δ) and a last-point-only control, all trained and evaluated under the main protocol. Capping the grid lowers mean FID from 3.291 to 1.458.

Supervision grid	FID↓
Original (uncapped)	3.291
Capped (TACD)	1.458
Last point only	1.462

Table 15: **HY-Motion 1B/$K{=}4$ supervision-grid control.** Three-seed mean FID; uniform segment weights. Rose row: terminal coefficient 10^6.

Capped ($\delta{=}1/(2K)$, $w_k{\equiv}1$): FID $1.418/1.417/1.540$, mean 1.458 (range 1.417–1.540); R@3 $0.735/0.734/0.732$; Div $8.25/8.26/8.21$. *Uncapped* ($\delta{=}10^{-3}$, $w_k{\equiv}1$): FID $3.233/3.305/3.335$, mean 3.291 (range 3.233–3.335); R@3 $0.689/0.684/0.686$; Div $7.73/7.72/7.70$. With $M{=}2$ supervision times per segment the cap moves both of the final segment's supervision times $(0.8745, 0.999 \to 0.8125, 0.875)$. The last-point-only control separates the two: with the grid built on the original knots and only the final supervision time clamped $(0.8745, 0.875)$, FID is 1.462 (3 seeds, $1.418/1.425/1.544$, range 1.418–1.544), so moving the last supervision time alone is sufficient in this configuration; it matches the capped arm seed for seed ($-0.001/{+}0.008/{+}0.004$).

I.2 COEFFICIENT INTERVENTIONS BY SEED

Table 16 gives, seed by seed, the two single-coefficient interventions of Section 4.3: lowering the terminal coefficient on the original grid, and restoring it on the capped grid.

Training configuration	Seed 42	Seed 43	Seed 44
Original, $C = 256$ (weight capping)	1.0437	1.0098	0.9798
Capped, $C = 10^6$	2.6593	2.6454	2.7029

Table 16: **Direct coefficient interventions by training seed.** Entries are FID; the first row is weight capping (❸) of Table 5, the second restores the terminal coefficient on the capped grid.

Weight capping (❸ in Table 5) is the first row of Table 16; its three-seed mean FID is 1.011. Restoring 10^6 on the capped grid gives 2.6593, 2.6454, and 2.7029 at seeds 42, 43, and 44 (mean 2.669).

I.3 GRADIENT PROBES AND CLIPPING

The gradient shares and norms of Section 4.3 come from the probes and training logs described here.

Full-backbone gradient probes. We compute the terminal segment's gradient norm divided by the sum of all segment gradient norms within each minibatch, then average those shares over 32 minibatches (a mean of ratios). At initialization and $K = 8$, the terminal segment carries 68.3% of the gradient on the original supervision grid and $0.86/0.89\%$ on the capped grid for probe seeds $42/43$. The eight final-checkpoint cells in Table 17 use those same probe seeds and 32 minibatches each, loading `full_ema` from the 8,000-iteration training-seed-42 checkpoint of each arm. All eight use the full 465.041M backbone.

Gradient magnitude is normalized by clipping. All Lite/$K = 8$ arms of Table 5 train with AdamW and gradient clipping at unit norm. Table 17 reports the pre-clip gradient norm, logged every ten updates. At the logged updates the clip is active in essentially every observation (all 800 in 17 of the 18 runs, 799 in the remaining one), so the compared configurations share the same clipping regime; the arms differ in how that gradient is composed across segments, which Table 17 measures as the terminal share.

I.4 TERMINAL-COEFFICIENT SWEEP

Table 18 lists the arms behind Fig. 6.

Final-segment share of gradient norm (%), seeds 42 / 43		
Grid probed	Uncapped ckpt.	TACD ckpt.
Original	96.62 / 96.64	97.60 / 97.10
Capped	10.53 / 11.77	4.28 / 3.64
Pre-clip gradient norm, training seeds 42 / 43 / 44		
Configuration	Median norm	Clipped (of 800)
Uncapped student	245.0 / 251.8 / 263.4	all
TACD (grid capping)	5.97 / 6.06 / 6.03	all
Endpoint matching	5.12 / 5.04 / 4.96	799 / 800 / 800
Weight capping	6.05 / 5.91 / 6.06	all
Last-point-only	5.92 / 6.10 / 6.05	all
Capped grid, $C = 10^6$	527.4 / 533.6 / 541.3	all

Table 17: **Gradient statistics on the eight-step Lite student.** Top: final-segment share of the summed segment-gradient norms, by probe grid and checkpoint (probe seeds 42/43). Bottom: median pre-clip gradient norm and the number of the 800 logged updates on which the unit-norm clip binds (training seeds 42/43/44).

Family	Coefficient	FID↓	R@1↑	R@3↑	MM-Dist↓	Div→	Norm
Weight family: original grid (queries .937, .999), terminal coefficient set to C by Eq. (15)							
Weight	$C = 256\ (4K^2)$	1.0111	.4800	.7814	3.190	8.925	6.01
Weight	$C = 1024$	1.0379	.4792	.7823	3.180	8.889	5.88
Weight	$C = 4096$	1.0671	.4802	.7823	3.172	8.879	6.24
Weight	$C = 16384$	1.1356	.4754	.7797	3.188	8.820	7.77
Weight	$C = 65536$	1.2995	.4707	.7769	3.221	8.640	17.82
Weight	$C = 262144$	1.6985	.4457	.7559	3.367	8.384	67.59
Weight	$C = 10^6$ (uncapped)	2.4859	.4024	.7076	3.703	8.005	253.4
Grid family: last query moved to $1 - \delta$, coefficient δ^{-2}							
Grid	$\delta = 1/32\ (b = 1024)$	0.9955	.4834	.7856	3.172	8.932	5.91
Grid	$\delta = 3/32\ (b = 113.8)$	1.0329	.4794	.7828	3.177	8.914	6.01
Grid	$\delta = 1/16\ (b = 256$, TACD)	1.041*	–	–	–	–	6.26

Table 18: **Terminal-coefficient sweep on Lite/K=8.** Each arm is the recipe of Table 5 with one flag changed (Guo evaluator, 20 repetitions). Norm: median pre-clip gradient norm over 800 logged updates. *Three-seed mean of Table 8 (range 1.009–1.074).

Each arm is the recipe of Table 5 with one flag changed: the weight family keeps the original grid (queries .937 and .999) and sets the terminal coefficient to C by Eq. (15); the grid family moves the last query to $1 - \delta$, which sets the coefficient to δ^{-2}. Every arm was scored with the Guo evaluator over 20 repetitions. The $C = 10^6$ and $C = 256$ arms are the uncapped and weight-capping students of Table 5. For $C \leq 4096$, and across the grid family, FID stays between 1.0 and 1.07, and it degrades progressively at larger coefficients. Above 4096 the five metrics move together at every step, and the median pre-clip norm rises from 6 to 253 while the unit-norm clip binds on all 800 logged updates of every arm: the coefficient changes the relative supervision weights, and the same unit-norm clipping rule is applied across configurations.

J Deployment cost

This appendix documents the latency and memory measurements behind Fig. 4 and Section 4.5 (Appendix J.1).

J.1 Latency and memory

The batch-1 cost numbers in the paper (Fig. 4 and Tables 12 to 13 and 19) come from one rig and one protocol: an RTX L40, `bf16` weights for the denoiser and the text encoder, batch 1, all weights resident, no disk I/O in the timed loop, and `cuda.synchronize` around text encoding and around sampling separately. HY-Motion rows generate a 196-frame (6.5 s at 30 fps) motion at CFG 5, Kimodo rows an eight-second motion. Each row runs 8 warm-up and 30 timed motions per pass, five passes in round-robin over all rows so that load drift on the shared machine is spread over every row, and the reported latency is the median of the 150 timings. Peak memory is the allocator's peak after loading, for text encoding plus one sample.

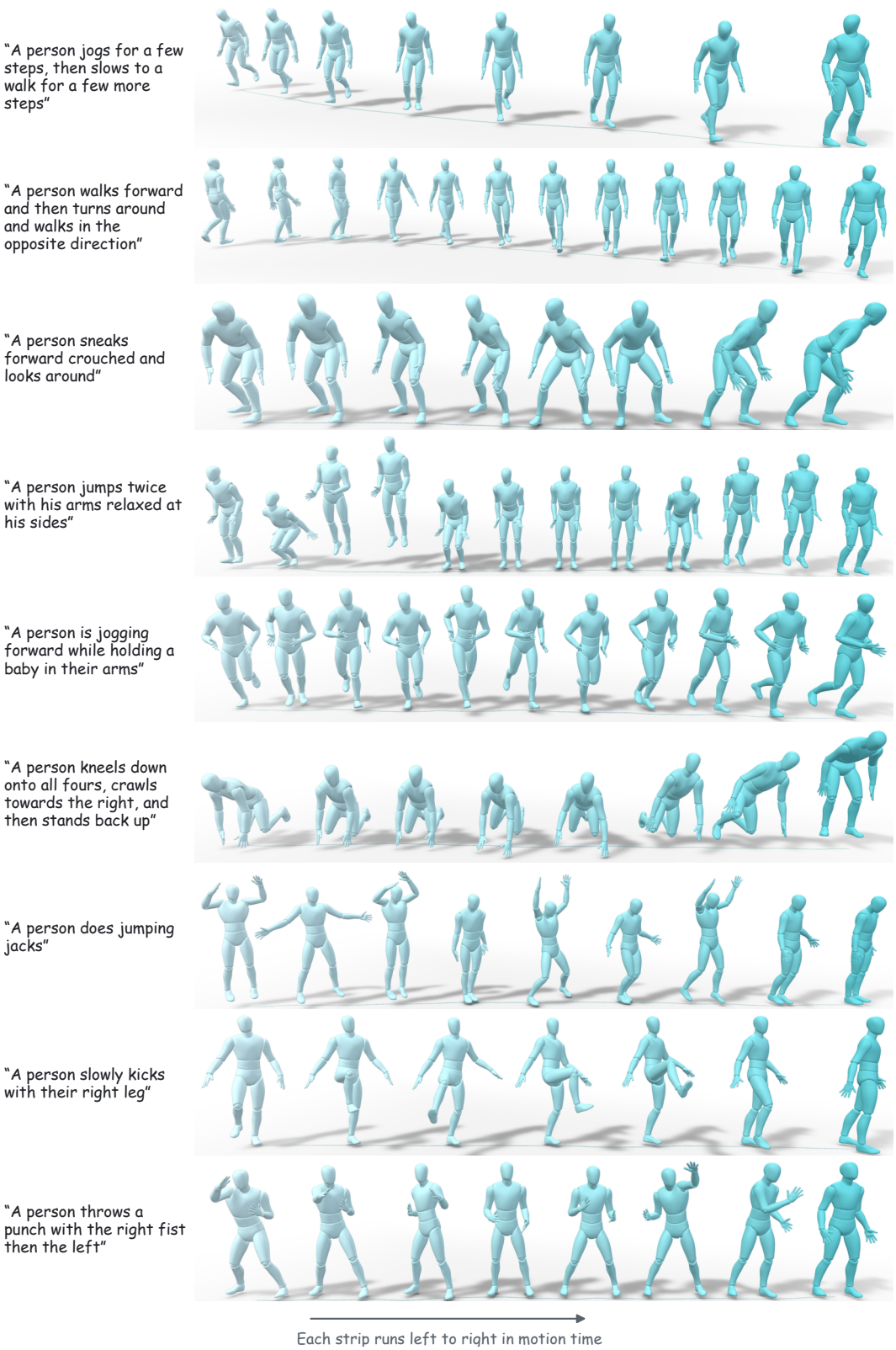


Figure 7: **Eight-step text-to-motion with TACD (I).** One prompt per row; each row is a single eight-step sample of TACD (HY-Motion), the Qwen3-0.6B/Lite student, shown at evenly spaced time points (Appendix K.2).

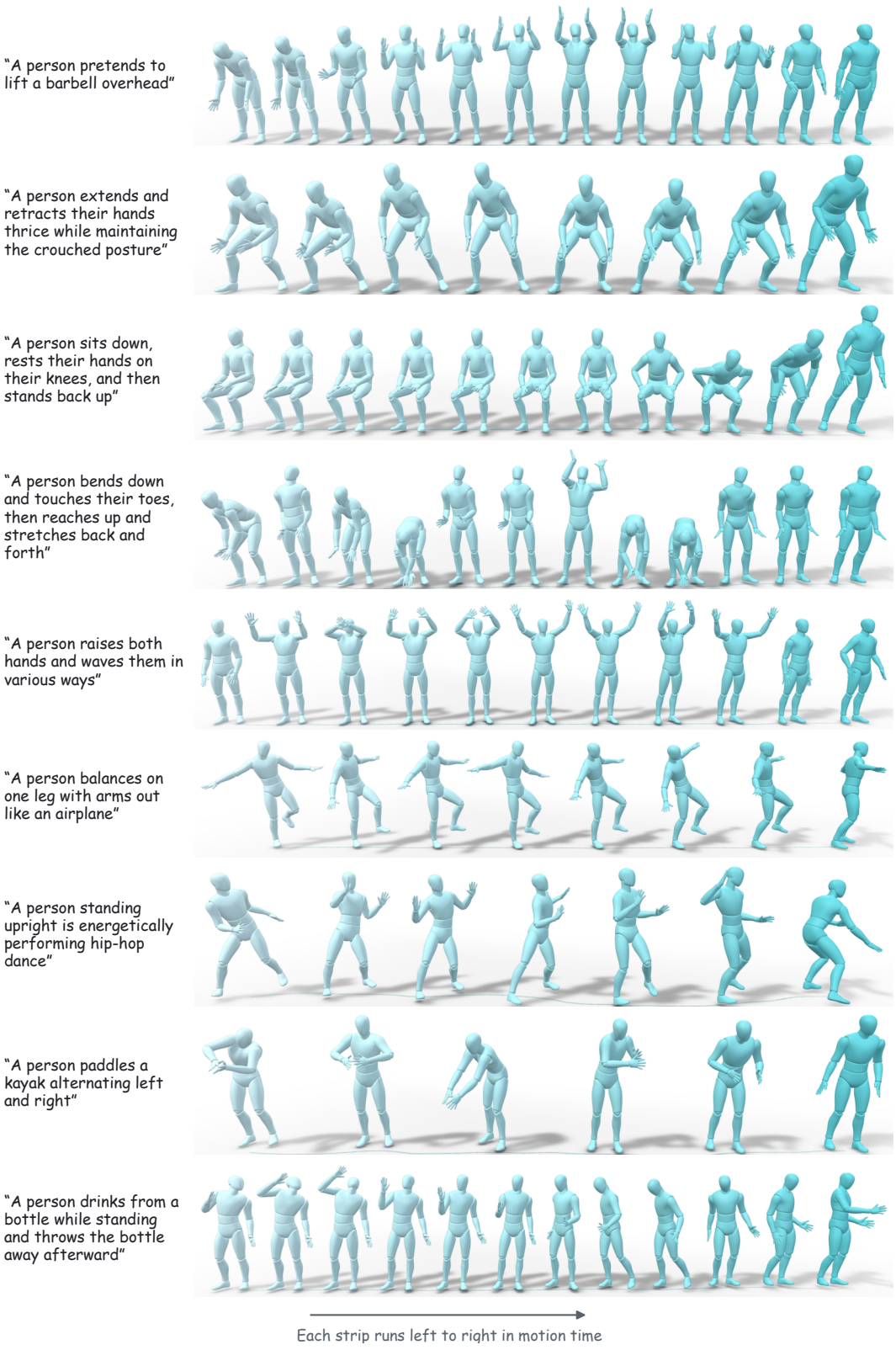


Figure 8: **Eight-step text-to-motion with TACD (II).** Continued from Fig. 7; same student, same rendering.

Table 19 lists the HY-Motion rows. Text encoding costs 41–54 ms per caption at batch 1, a small portion of end-to-end latency. Sampling tracks the step count: TACD (HY-Motion) at eight steps samples in 227 ms vs. 1422 ms for 50-step Lite (6.3× for a 6.25× step reduction). End to end, TACD (HY-Motion) is 5.5× faster than 50-step Lite and 7.7× faster than the teacher, at 2.62 GB against the teacher's 17.63 GB (6.7×); the saving is the text encoder (8B to 0.6B) and the backbone (1B to 0.46B), and the student adds only the 1024→4096 conditioning projection to the released Lite architecture.

Model / steps	end to end	sampling	text enc.	peak GB
1B teacher + Qwen3-8B, 50 steps	2077 ms	2026	54	17.63
0.46B Lite + Qwen3-8B, 50 steps	1476 ms	1422	52	16.54
TACD (HY-Motion): 0.46B + Qwen3-0.6B, 8 steps	268 ms	227	41	2.62

Table 19: **Batch-1 cost of the HY-Motion systems** on one RTX L40 (`bf16`; medians over five round-robin passes of 30 motions of 6.5 s). Text encoding is per caption.

Training cost. Per iteration, on-policy supervision adds one K-step rollout of the rollout-EMA copy, K gradient-carrying student evaluations at the detached segment starts, and KM frozen-teacher queries. All of this is training-only: inference NFE is unchanged, and the EMA copy does not affect deployment.

K Qualitative results

This appendix shows generated motions: the teacher, its eight-step truncation and TACD side by side (Appendix K.1), and a gallery of eight-step samples of the compact student (Appendix K.2).

K.1 Teacher, truncated teacher, and TACD

Fig. 1 shows four prompts, each rendered for the 50-step teacher, for the same teacher truncated to the eight-step sampling grid, and for TACD at eight steps. Within a row the prompt, the initial noise, the frame indices and the camera are shared: the three cells of a row come from a single render whose camera is fitted to the union of the three motions.

K.2 Eight-step gallery

Figs. 7 and 8 show one prompt per row. Each row is one eight-step sample of the TACD (HY-Motion), one draw per prompt. Within a row the poses are evenly spaced time points of that one clip. For visualization only, each clip is rotated by a fixed yaw so that the body faces the reader; this does not alter the motion.