MinPair: Measured Minimal Pairs for Fine-Grained Control in Text-to-Motion Generation

Lennart Schiweck Cristóbal Curio
Cognitive Systems Group, Reutlingen University, Germany

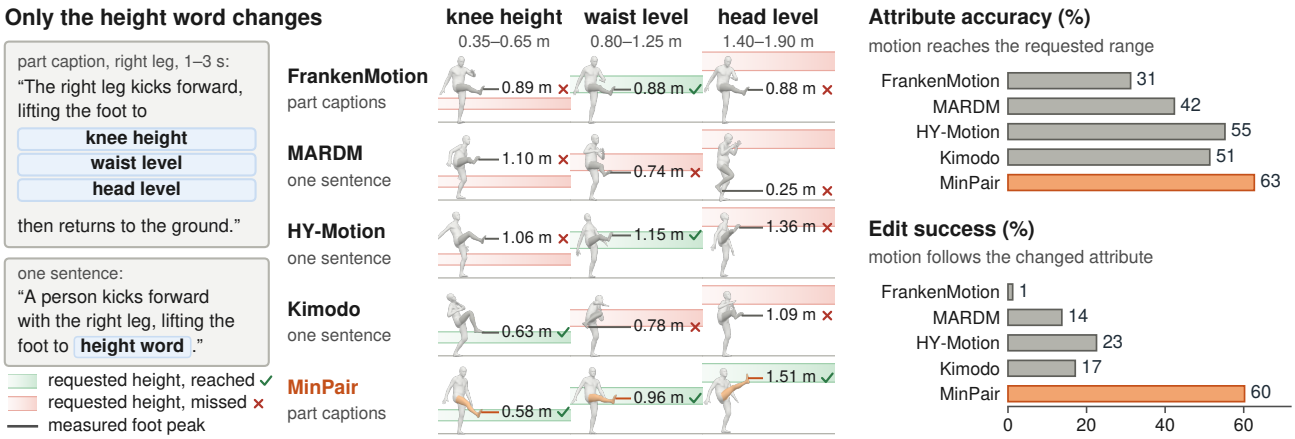


Figure 1. **Kick height requests.** Only the height word changes. Like FrankenMotion [21], MinPair reads the body-part caption of the right leg, whereas MARDM [24], HY-Motion [37] and Kimodo [31] read one sentence. Shown per model and height is the motion closest to the median foot peak of all motions generated from that prompt, with the requested range in green if the foot peak lies inside it and in red if it misses. The range below each height is the benchmark's fixed target range. MARDM, HY-Motion and Kimodo are SMPL-H fits to their stored joints. Only MinPair reaches all three ranges. Right, attribute accuracy over all 22 target tasks and the share of template edits followed, each with the input of its row (Sec. 4.1).

Abstract

Text-to-motion generation has made great progress in capturing what action to perform, yet it remains limited in controlling how that action is executed. For example, when only the height word of a kick request changes, none of the tested state-of-the-art methods reaches all three of knee, waist and head height. The other words of a caption usually imply an attribute such as height, so neither the training loss nor the standard metrics require a model to use the word that states it. A model has to use this word only when it alone separates two otherwise identical captions. We introduce MinPair, which creates such caption pairs from real motion clips. It starts from rule-based motion descriptors that measure the 3D body motion of each clip. From these descriptors a language model writes MinPair captions, timed body-part captions that say how each body part moves. The same descriptors show in which attributes two clips of the same action differ. A measured minimal pair joins the true caption of one clip with an altered copy in which only one attribute takes the value measured in the other clip. A contrastive pair loss then requires a diffusion denoiser to reconstruct the clip more accurately from its true caption than from the altered copy. Training needs no synthetic motion and inference stays unchanged. Control is tested on 22 target tasks and on caption edits that each change one attribute. MinPair captions alone help motions reach their targets and improve the standard metrics M2M, R@3 and FID, yet a model trained on them still ignores most single-word edits. The pair loss raises the share of followed template edits from 11% to 60%, while the standard metrics do not improve further. With body-part captions, MinPair reaches more targets than state-of-the-art methods that were trained on up to 99 times more motion data.

1. Introduction

In text-to-motion generation [8], a request such as *kick to head level* should produce a kick that reaches the head. The action alone leaves open how high, how fast and how far a

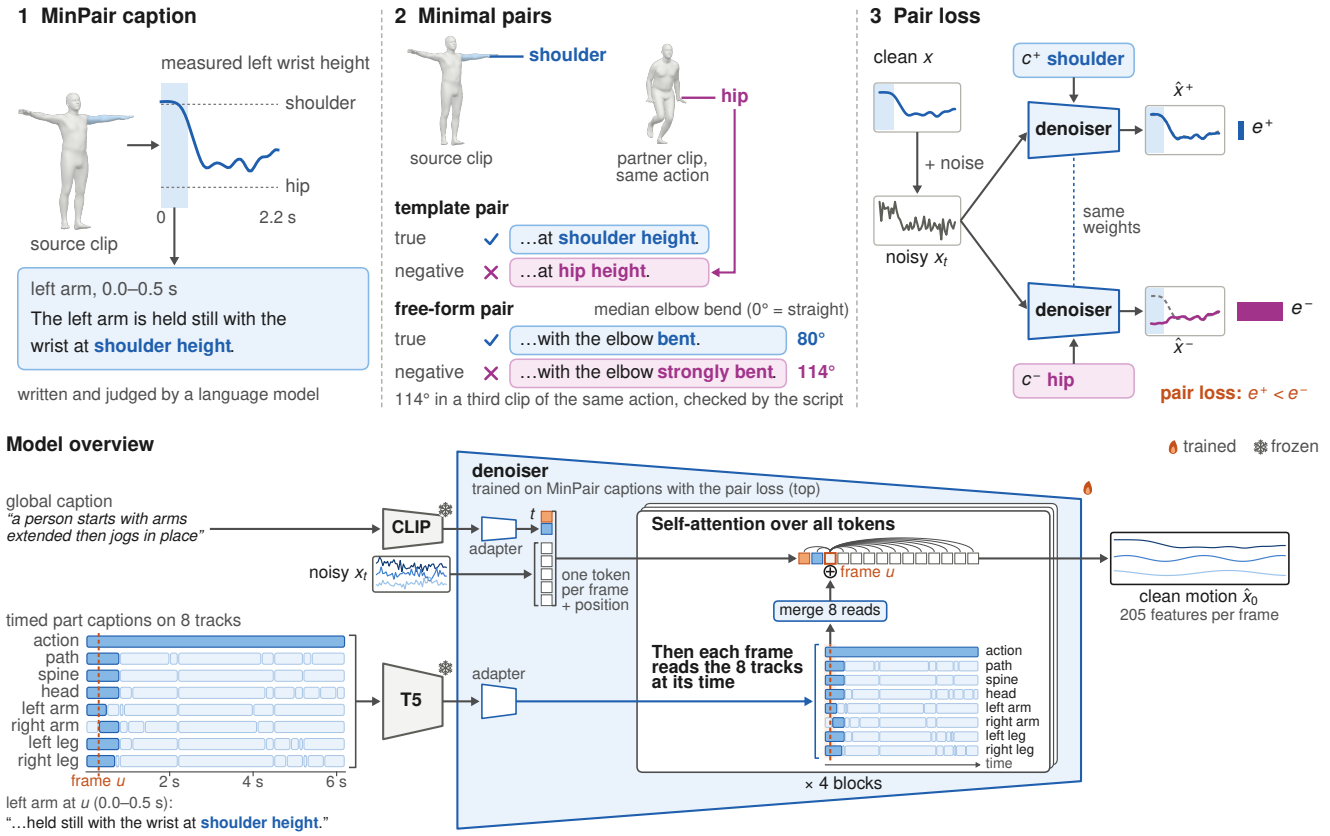


Figure 2. **Method overview.** Top, training. (1) A script measures quantities in a real clip, here the left wrist height. A language model writes them into a timed part caption and a further call judges it against the measurements. (2) A template pair copies the true caption with the value of a partner clip of the same action, here hip height. A free-form pair rewords a short span after a measured difference such as the elbow bend and is checked against the measurements. (3) Both captions condition the same denoiser on the same noisy motion. The pair loss asks the true caption for the smaller error inside the caption's time window (shaded). The curves in (3) are illustrative. Bottom, the part-track denoiser of Sec. 3.1, shown for frame u. Only the text encoders are frozen.

movement goes and how much it turns. We call such execution properties *attributes*. An animator who asks for a head-level kick or a robot operator who asks for *three steps forward* expects exactly that execution. Diffusion models have become the standard generator for this task [11, 35]. FrankenMotion [21] gives every body part its own timed track of sentences, which we call *part captions*. HY-Motion [37] and Kimodo [31] scale data and parameters. Yet when only the height word of a kick request changes, none of these state-of-the-art methods reaches all three requested heights (Fig. 1). Nothing in their training singles out the attribute word. Text labels [8, 27] rarely state how high, how fast or how far a movement goes, so part captions derived from them [21] rarely do either. Where captions do state it, the other words give it away. In captions written from measurements, a linear classifier recovers a masked tempo word from the remaining words of a held-out caption in 77% of cases against 41% for the most frequent tempo. The denoising loss and the standard metrics M2M, R@3 and FID all reward the complete caption, so a model can succeed from the context without using the attribute word. A model learns to follow a word reliably only once that word alone decides the outcome.

A motion corpus already records many executions of each action, such as kicks to knee and to head height. Measuring them shows which words fit each execution [4, 22, 40] and which clips of the same action differ in a given attribute. Measured minimal pairs create these cases from real clips.

We introduce MinPair (Fig. 2). It measures every training clip with rule-based motion descriptors. From these a language model writes *MinPair captions*, which say how each body part moves in each phase. For a source clip whose caption places the wrist at shoulder height, a partner clip of the same action keeps it at hip height. We copy the source caption and change only this value to the partner's. The copy is wrong for the source clip in exactly one attribute and serves as its *negative caption* [41]. The new words arise

in one of two ways. In a *template pair*, the attribute word moves along a fixed word scale of the captions. In a *free-form pair*, a language model words the measured difference itself in one span of up to 4 words, which we check against the measurements. Because the partner is a real clip, no motion is synthesized and the negative describes an execution that really occurs. As in contrastive learning [36], a *pair loss* then requires the denoiser to reconstruct the same noisy copy of the source motion more accurately from the true caption than from the negative [17, 19]. Only the changed words can make this difference. Inference stays unchanged.

We test control with two benchmarks (Sec. 4.1). *Target tasks* request explicit values such as a kick to waist level or a walk of three steps. *Caption edits* change one attribute in the MinPair captions of held-out clips, as template or free-form edits built like the two kinds of pairs. With MinPair captions but without the pair loss, our denoiser follows only 11% of the template edits. The pair loss raises this share to 60% while M2M, R@3 and FID do not improve further. With body-part captions MinPair reaches more targets than state-of-the-art methods [21, 24, 31, 37], although HY-Motion and Kimodo were trained on 23 to 99 times more motion data (Sec. 4.2).

Our contributions are as follows.

- We introduce measured minimal pairs and a contrastive pair loss that make a single attribute of a caption controllable in text-to-motion diffusion models. Each pair combines the true caption of a clip with a negative caption that takes one value from another real clip of the same action, showing a targeted form of supervision that can outweigh scale (Secs. 3.3, 3.4 and 4.2).
- We introduce MinPair captions, timed part captions written from measurements of each motion. They state the attributes of each body part in each phase. With the same network they improve attribute accuracy and all three standard metrics over previous part captions (Secs. 3.2 and 4.3).
- We introduce two control benchmarks, target tasks and caption edits, which isolate single attribute words. They show that controlling a single attribute is a separate skill from fitting a complete caption, which M2M, R@3 and FID do not measure (Secs. 4.1 and 4.3).

2. Related Work

Text-to-motion generation and fine-grained text. HumanML3D [8] became the standard benchmark for text-to-motion generation and MDM [35] made denoising diffusion the common generator. Recent work refines the text side. FineMoGen [42] generates motion from fine-grained spatio-temporal descriptions, STMC [26] composes timed prompts on a multi-track timeline during sampling and FrankenMotion [21] conditions each body part on its own timed sentences. FineMotion [38] and KinMo [43] describe the movements of single body parts or kinematic groups. FineXtrol [32] and MATE [9] learn finer text representations with hierarchical or word-level contrastive objectives. Other work scales the generator. HY-Motion 1.0 [37] trains a billion-parameter flow-matching transformer on more than 3,000 h of motion, Kimodo [31] trains a diffusion model on a large optical motion-capture corpus, OpenT2M [2] scales data with detailed temporal annotation and MotionMaster [14] unifies generation and editing. PoseScript [4], PoseFix [5], Kinematic Phrases [22] and MotionScript [40] connect measured pose and motion properties to language and CoMo [13] generates and edits motion through pose codes of single body parts. LaMoGen [16] guides a pretrained diffusion model with the effort and shape qualities of Laban Movement Analysis [18]. MinPair builds on measured part captions and adds a training signal that makes the generator separate attribute values within one action.

Hard negatives and post-training. ARO [41] uses composition-aware negative captions to expose weaknesses of vision-language models and SugarCrepe [12] shows that carelessly built negatives contain linguistic shortcuts. Diffusion Classifier [20] classifies an image by the text that gives the lowest denoising error. HardNeg-DiffusionITM [17] and SoftREPA [19] train with contrastive objectives over denoising errors and TDSM [6] recognizes skeleton actions with a triplet loss over them. Fujiwara *et al*. [7] train motion-language models with negative texts whose events are shuffled in time. MinPair transfers the denoising-based principle to motion generation and mines its negatives from measured differences between two real executions of one action. MoDiPO [25] adapts diffusion preference optimization to motion and Motion-RFT [34] applies reinforcement fine-tuning with learned motion rewards. OmniME [33] learns motion editing from positive and negative examples. MinPair needs neither a reward model nor generated candidates. It contrasts two captions of one real motion.

Evaluating instruction following. SATO [3] tests stability under semantically similar texts. T2MBench [39] evaluates instruction following, physical plausibility and fine control across many prompts and models, with numeric targets for the whole body and for pairs of body parts. MotionFix [1] benchmarks motion editing with triplets of source motion, target motion and edit text. Our benchmarks add target ranges for single body parts and paired caption edits under identical sampling noise, which isolate the effect of one changed attribute.

3. Method

MinPair measures every training motion, writes the measurements into timed part captions and pairs these captions with negative captions that each change one measured attribute (Fig. 2).

3.1. Part-track diffusion model

As in FrankenMotion [21], a caption consists of a global caption for the whole clip and eight tracks for action, path (the root trajectory), spine, head, left arm, right arm, left leg and right leg. We call one sentence with its time span a *segment*. Tracks need not share their boundaries.

A motion clip $x \in \mathbb{R}^{T \times D}$ has T frames with $D = 205$ features each, in the motion representation of FrankenMotion. Its text condition c is such a part caption. MinPair is a diffusion model [11, 35]. The forward process mixes x with Gaussian noise,

$$x_t = \sqrt{\bar{\alpha}_t}\, x + \sqrt{1 - \bar{\alpha}_t}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \qquad (1)$$

for a diffusion step $t \in \{1, \ldots, 100\}$ with a cosine schedule $\bar{\alpha}_t$. We call x the clean motion and x_t the noisy motion. The denoiser $f_\theta(x_t, t, c)$ predicts the clean motion from the noisy one. For a set W of frame indices, such as the time span of one segment, its reconstruction error under caption c is

$$e_W(c) = \frac{1}{|W|} \sum_{u \in W} \left\| f_\theta(x_t, t, c)_u - x_u \right\|_\omega^2, \qquad (2)$$

where ω weights the motion features as in FrankenMotion. The error covers all features of the frames in W, not only the joints of one body part. The denoising loss $\mathcal{L}_{\text{denoise}} = \mathbb{E}_{(x,c),\, t,\, \epsilon}[e_M(c)]$ averages this error over the set M of all valid frames of a clip with t drawn uniformly.

Part-track denoiser. The denoiser is a Transformer over frames (Fig. 2, bottom). Every frame of x_t becomes one token. Two more tokens carry the diffusion step t and the global caption, which frozen CLIP [29] encodes. Frozen T5-Large [30] encodes every segment. Trained adapters map both text features into the denoiser. Each of four blocks first applies self-attention over all tokens followed by a feed-forward layer. Then every frame reads the segment that is active at this frame on each of the eight tracks through cross-attention. A small MLP fuses the eight reads with flags that mark the active tracks and adds the result to the frame token. A linear head predicts the clean motion.

3.2. MinPair captions

MinPair writes the seven body-part tracks of almost every caption from the motion itself. The global caption and the action track remain those of FrankenMotion [21], which a language model merged from existing text labels, mostly from BABEL [27] and HumanML3D [8].

First, we compute the joint positions of every clip with the body model and shape of its recording. Rule-based motion descriptors summarize them in a short text that states every 0.75 to 1 s which limbs move, how high hands and feet rise, how the joints bend and how far the pelvis travels and turns. Qwen3.6-27B [28] reads this summary with a table of how much each body region moves and cuts the clip into phases of the whole body. It sees no text labels in this step. For every phase and body part we then turn the measured values into level words on fixed word scales, such as *knee height*, *shoulder height* or *overhead* for heights and *still*, *slow*, *moderate* or *fast* for tempo. The distance travelled becomes a number of steps and the turn an angle rounded to 5°.

A second call of the language model writes the sentences from the summary, the phases and these level words. The prompt asks for one sentence per body part and time span and for tracks without gaps. It asks to describe a repeated movement such as walking in one segment that states its count, to name the side of every arm and leg and to give heights and directions relative to the body, as in *across the body*. It allows numbers only for turn angles and counts. The call also sees the BABEL and HumanML3D labels of the clip as hypotheses. It may name the activity after them only where the measurements support them for that body part and time span. A third call judges every sentence against the summary. Its verdict is stored for inspection and removes no sentence. The caption in Fig. 2 (1) arises this way from the wrist-height curve.

MinPair captions comprise 450,179 segments on 12,726 training clips, a median of 29 per clip. The same pipeline writes MinPair test captions for 1,583 test clips, which no model sees in training.

3.3. Measured minimal pairs

A minimal pair consists of two captions of the same clip that differ only in the words of one attribute. The *negative caption* copies the true caption of the source clip and changes only these words to the value of a partner clip, another real execution of the same action. All other words, segments and tracks stay the same. The partner only supplies this value and is never reconstructed. Template pairs and free-form pairs differ in how the new words arise.

Template pairs. In a template pair, the attribute word moves along a fixed word scale of the captions. We search for two training clips of the same action whose captions state different values of one attribute on the same track, such as the highest height word of a leg track, a tempo word, a turn angle or a step count. A value counts only when the descriptors of its clip confirm it. The partner's word then replaces the source word, for example *hip height* for *shoulder height* (Fig. 2 (2)). A second search compares two clips of

the same action directly on 51 measured quantities. There a difference must exceed the measurement noise and a visibility threshold, such as half a height level of the person's body or 30° of turn. Template pairs cover height, tempo, turn, count and side and the four posture attributes trunk lean, trunk twist, head speed and knee bend.

Free-form pairs. In a free-form pair, the language model words the measured difference itself in one span of up to 4 words. This also covers differences that the word scales of the captions do not name, such as elbow bend or stance width. For each training clip we pick up to 2 partners of the same action and compare both clips on 61 measured quantities. The language model receives the source caption with its time spans and, for every visible difference, the descriptor values of both clips with their words. It returns a true and a negative sentence for each pair, where the negative changes one span to the partner's value. The true sentence may add the source value to a segment that does not state it and then also replaces this segment in the denoising loss. We check that the sentence lies on a track and time span where the difference occurs and that the changed words point from the source to the partner value wherever the check can read both values from the words. A rejected pair returns once to the language model with the reason. Of the proposed pairs 72.4% pass.

Training uses 190,141 negative captions for 11,824 of the 12,822 training clips, 48,155 of them from free-form pairs.

3.4. Pair loss

For a negative caption c_j^- with affected time window W_j, we compare the reconstruction errors of the same noisy motion under both captions,

$$z_j = \frac{e_{W_j}(c^+) - e_{W_j}(c_j^-)}{\tau \max\{\mathrm{sg}[e_{W_j}(c^+)], 10^{-8}\}}, \tag{3}$$

where c^+ is the true caption or its paraphrase, sg stops the gradient and $\tau = 0.1$. For a clip with J negative captions the pair loss is

$$\mathcal{L}_{\mathrm{pair}} = \log\Big(1 + \sum_{j=1}^{J} \exp z_j\Big). \tag{4}$$

Dividing by the detached positive error makes the difference relative, so clips and diffusion steps with large errors do not dominate the loss.

Relation to contrastive learning. Define the score of a caption as $s_W(c) = -e_W(c)/(\tau\ \mathrm{sg}[e_W(c^+)])$. Then $z_j = s_{W_j}(c_j^-) - s_{W_j}(c^+)$. For negatives that share one window W, Eq. (4) is exactly the InfoNCE loss [36] with these scores over one positive and J negative captions. Training is thus contrastive learning over captions in which the denoiser is the scoring function, as in HardNeg-DiffusionITM [17] and SoftREPA [19] for images. MinPair mines its negatives from measurements and restricts each comparison to the window of the changed segment.

The objective adds the pair loss to the denoising loss,

$$\mathcal{L} = \mathcal{L}_{\mathrm{denoise}} + \lambda\, \mathbb{E}_{\mathrm{sel}}\big[\mathcal{L}_{\mathrm{pair}}\big], \quad \lambda = 0.05, \tag{5}$$

where the pair loss is averaged over the selected clips of a batch and the denoising loss covers all clips.

3.5. Training

We train the denoiser from random initialization with Eq. (5) for 1,500 epochs (300,000 steps at batch size 64). We use AdamW with a constant learning rate of 10^{-4}. An exponential moving average of the weights (decay 0.9995) gives the evaluated model. A clip with negative captions is always eligible for the pair loss at the noisier steps $t \geq 50$ and with probability 0.2 at the less noisy steps below. Each batch selects at most 21 of its 64 clips, those with negatives from a partner first. For template pairs, a paraphrase of the true segment such as *shoulder level* for *shoulder height* replaces it with probability 0.5 while the negative keeps its wording, so new wording does not signal a negative [15]. A clip without a partner receives up to 2 vocabulary swaps, negatives in which another word of the same scale replaces the stated value. Two further negatives join each with probability 0.5. One exchanges left and right in all texts and tracks, the other holds no text at all. Training draws clips with a rare high foot or wrist position up to 2.95 times more often. It takes about 21 h on one H100 GPU.

4. Experiments

4.1. Setup

Data. We use the AMASS [23] subset of FrankenMotion [21] with its split into 12,845 training clips of 30 h in total, 1,651 validation and 1,601 test clips. Motions run at 20 fps. We keep clips between 1 and 20 s, which leaves 12,822 training and 1,596 test clips.

Baselines. We compare MinPair with four published generators. FrankenMotion [21] is the part-track model whose input format MinPair shares. We use its released checkpoint, which is trained on the same AMASS split of 30 h. Both part-track models use the same sampler without classifier-free guidance [10]. MARDM [24], HY-Motion 1.0 [37] and Kimodo [31] read a single text. MARDM is a masked autoregressive diffusion model trained on HumanML3D [8]. HY-Motion and Kimodo were trained on more than 3,000 h and on 700 h of motion. We run the official weights with the sampling settings of each

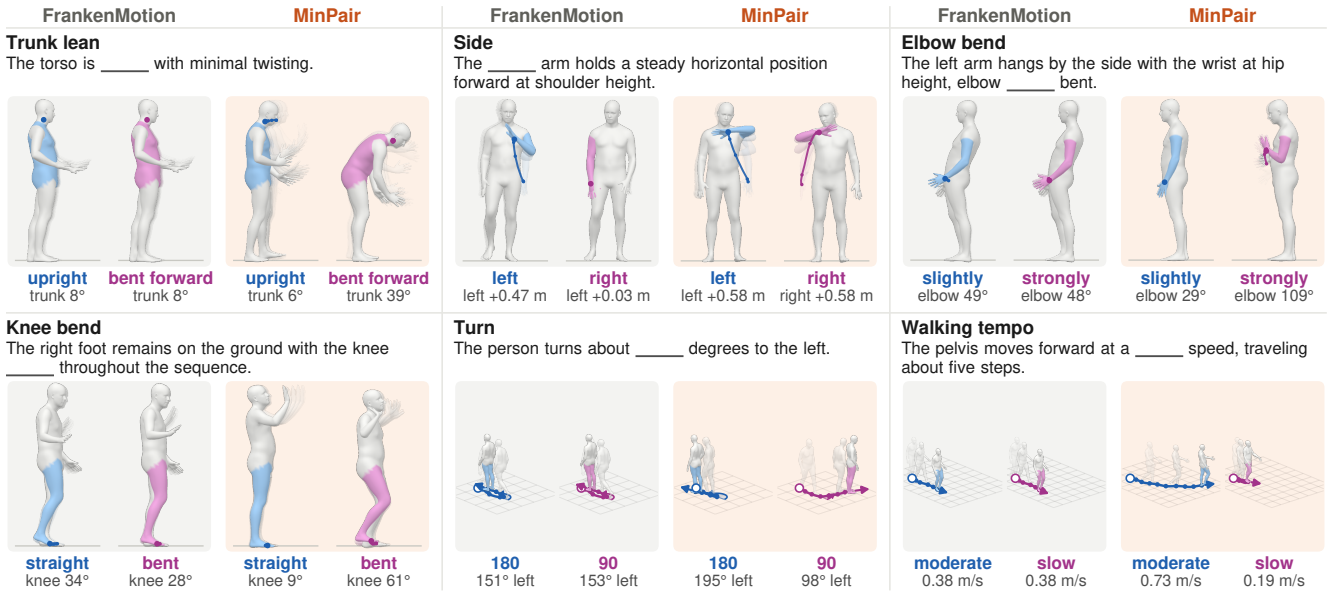


Figure 3. **Caption edits.** Each block changes one attribute in the part captions of a held-out test clip (blue to magenta) under the same sampling noise. Elbow and knee are free-form edits, the others template edits. Turn and tempo show the pelvis path, the numbers the measured value. Over all test edits of these six kinds MinPair follows 83, 49, 72, 53, 48 and 43%, FrankenMotion at most 3%. Each block shows a clean edit with a large requested change whose motions come close to the requested values.

release and resample the joints to 20 fps. MARDM generates at most 9.8 s and HY-Motion at most 12 s. Kimodo generates every clip in one prompt at its length.

Inputs. Every model reads the input it was designed for. FrankenMotion and MinPair read *part captions*, so a request is written into the timed segment of the relevant body part in the sentence style of MinPair captions. MARDM, HY-Motion and Kimodo read a single text, so they receive the whole request as *one sentence*.

Target tasks. The first benchmark tests whether a model reaches a requested value in 22 tasks such as kick height, arm height, turn angle, walking distance, squat count, trunk lean and knee bend. Each task has three requested levels such as knee, waist or head height. Every level is written in both inputs and generated with 20 seeds shared across models. A script m_q measures the quantity q of the task, such as the peak foot height, in the motion $G(c_\ell, \xi)$ generated from the prompt c_ℓ of level ℓ with sampling noise ξ. A sample is a hit when this value falls into the target range R_ℓ of the requested level,

$$\mathrm{Hit}_A = \mathbf{1}\big[m_q(G(c_\ell, \xi)) \in R_\ell\big]. \tag{6}$$

Attribute accuracy is the share of hits on the target tasks, averaged over the task and level cells of an input.

Caption edits. The second benchmark tests whether a single changed word changes the motion. A caption edit changes only text. It replaces the value of one attribute in the MinPair captions of a held-out test clip. The model then generates one motion from the unedited caption c and one from the edited caption c' with the same noise ξ, so a difference between the two motions comes from the text alone. The edits come from two sources. *Template edits* are built by a script. They change arm height, leg height, tempo, turn, count or side in 3,292 edits of 1,574 clips and trunk lean, trunk twist, head speed or knee bend in 2,577 further edits by the rules of the template pairs. A side edit mirrors the left and right tracks. *Free-form edits* come from the language model of the free-form pairs, which changes one attribute of 18 kinds in 4,007 edits. The script checks them like the free-form pairs. Let $\Delta_g = m_q(G(c', \xi)) - m_q(G(c, \xi))$ be the generated change of the edited quantity q and Δ_r the change observed between real clips with the two attribute values. An edit counts as *followed* when

$$\mathrm{Hit}_B = \mathbf{1}\big[\mathrm{sign}(\Delta_r)\,\Delta_g \geq \tfrac{1}{2}|\Delta_r|\big]\,. \tag{7}$$

The motion must move in the requested direction by at least half the real change. For the four posture attributes and the free-form edits Δ_r is the measured difference to one training partner, over segments for posture edits and over whole clips for free-form edits. Edit success is the share of followed edits of one source, where every edit counts once since some kinds hold only a few edits. Figure 3 shows

Table 1. **Comparison with published generators.** Each model reads its own input format. One-sentence inputs carry no side edits. Success and R@3 in %, the best value of each column in bold.

Model	Input	Data (h)	Target tasks ↑	Template edits ↑	Free-form edits ↑	All edits ↑	M2M ↑	R@3 ↑	FID ↓
MARDM [24]	one sentence	<29	42.3	13.7	7.0	10.8	0.694	81.4	0.117
HY-Motion 1.0 [37]	one sentence	>3,000	55.2	22.5	16.7	20.0	0.754	**89.0**	0.059
Kimodo [31]	one sentence	700	51.4	17.1	13.4	15.5	0.618	56.7	0.431
FrankenMotion [21]	part captions	30	31.2	1.2	0.2	0.8	0.745	86.0	0.057
MinPair (ours)	part captions	30	**62.7**	**60.2**	**55.0**	**58.1**	**0.830**	85.9	**0.038**

Table 2. **Effect of captions and pair loss.** Every row trains our denoiser, also the first, which uses the FrankenMotion captions and the FrankenMotion training procedure. Each row changes one setting of the row above it. The last row is MinPair. Success and R@3 in %.

Our denoiser trained with			Target	Template	Free-form	All			
Captions	Training	Pair loss	tasks ↑	edits ↑	edits ↑	edits ↑	M2M ↑	R@3 ↑	FID ↓
FrankenMotion	FrankenMotion	–	27.5	1.4	0.1	0.9	0.729	81.6	0.089
MinPair	FrankenMotion	–	50.0	11.4	6.0	9.2	0.819	86.9	0.061
MinPair	MinPair	–	44.5	10.5	4.7	8.2	**0.837**	**89.0**	**0.033**
MinPair	MinPair	template pairs	60.7	**61.4**	27.6	47.7	0.834	88.0	0.034
MinPair	MinPair	free-form pairs	57.0	36.7	**59.8**	46.0	0.827	86.6	0.036
MinPair	MinPair	both	**62.7**	60.2	55.0	**58.1**	0.830	85.9	0.038

template edits of trunk lean, side, turn and walking tempo as well as free-form edits of elbow and knee bend.

Quality and alignment. We also report three metrics of the FrankenMotion evaluator [21] on the 1,562 test clips with a global caption, averaged over three generation runs. M2M is the rescaled cosine $(1 + \cos)/2$ between embeddings of the generated and the reference motion. R@3 is the top-3 text retrieval accuracy. FID is the Fréchet distance between embedding distributions. Each motion is generated from the complete caption of a real test clip and compared with that clip. Models trained on the FrankenMotion captions receive the FrankenMotion test captions and models trained on MinPair captions the held-back MinPair captions of the same clips, which the pipeline of Sec. 3.2 writes from the measurements of each test clip. The evaluator was trained on the FrankenMotion captions and scores every text metric against them. We therefore omit its motion-to-text similarity M2T and use M2M as our primary quality metric because it compares motion with motion. The three single-text models receive the global caption and the clip length.

Control runs. To separate the captions from the network, we also train our denoiser with the FrankenMotion training procedure and the denoising loss alone, once on the part captions of FrankenMotion and once on MinPair captions. Both runs share clips, split, settings and epochs and differ only in the captions.

4.2. Comparison with published generators

Table 1 gives every model its own input. With part captions MinPair reaches 62.7% attribute accuracy and follows 60.2% of the template edits and 55.0% of the free-form edits. This is more than any published generator reaches, although HY-Motion and Kimodo were trained on 23 to 99 times more motion data. When both part-track models instead read the one sentence of MARDM, HY-Motion and Kimodo in the global caption and in all eight tracks, FrankenMotion follows 10.0% of the template edits and 5.0% of the free-form edits. MinPair still follows 50.7% and 50.0%, more than any published generator. On the test clips of at most 9.8 s, within the length limit of every model, MinPair follows 58.2% of the edits against 0.9 to 19.7% for the published generators.

4.3. Effect of captions and pair loss

Captions and pair loss. Table 2 trains the same denoiser in every row and changes one setting per row. Trained on the FrankenMotion captions, this denoiser is no better than FrankenMotion itself (Tab. 1). Its attribute accuracy is 27.5% against 31.2% and it follows 1.4% of the template edits against 1.2%. The network therefore does not explain the gains that follow. MinPair captions alone raise attribute accuracy to 50.0% and template-edit success to 11.4%. The training of Sec. 3.5 without the pair loss adds no control (44.5% and 10.5%). The pair loss with template pairs then raises template-edit success to 61.4% and attribute accuracy to 60.7%. MinPair captions thus teach the model which ex-

Table 3. **Ablation of the pair loss training.** Each row below MinPair changes one part of its training and trains from scratch. Success and R@3 in %.

Model	Target tasks ↑	Template edits ↑	Free-form edits ↑	All edits ↑	M2M ↑	R@3 ↑	FID ↓
MinPair	**62.7**	**60.2**	55.0	**58.1**	0.830	85.9	0.038
pair loss at high noise only	57.6	49.4	44.3	47.3	0.829	87.0	**0.036**
pair loss at every step	57.7	59.4	**56.0**	58.0	**0.831**	**87.3**	**0.036**
w/o vocabulary swaps	58.3	58.5	53.9	56.6	0.827	85.9	0.039
w/o rare-position oversampling	61.9	57.9	53.0	55.9	0.830	86.9	0.038

ecution a complete caption describes. The pair loss makes it respond to a single changed word.

Pair sources. The last three rows of Tab. 2 differ only in their negative captions. Each source helps most on the edits it covers. Template pairs follow 61.4% of the template edits but only 27.6% of the free-form edits. Free-form pairs alone follow 59.8% of the free-form edits but only 36.7% of the template edits. MinPair combines both sources and follows 60.2% of the template edits and 55.0% of the free-form edits. Each source thus leads on its own edits, while MinPair, trained on both, leads in attribute accuracy with 62.7% and on all edits together with 58.1%.

Standard metrics. The standard metrics follow the captions but not the pair loss. Sequence M2M rises from 0.729 to 0.819 with MinPair captions and to 0.837 with the training of Sec. 3.5. With the pair loss it is 0.830 although template-edit success rises from 10.5% to 60.2%. The reason is that these metrics test complete captions. A complete caption describes an execution redundantly. When we mask the arm-height, leg-height or tempo word of a held-out caption, a linear classifier fit on the training captions recovers it from the remaining words in 86%, 73% and 77% of cases against 63%, 60% and 41% for the most frequent value. A model can therefore produce the right execution from the context without using the attribute word. Motions generated from the unedited test captions indeed agree with the real clip equally often with and without the pair loss, in 87.8% of cases over 8 measured quantities.

4.4. Ablation of the pair loss training

Table 3 changes one part of the pair loss training per row and trains from scratch. With the pair loss at high noise only, at the steps $t \geq 50$, the model follows 49.4% of the template edits and 44.3% of the free-form edits instead of 60.2% and 55.0%. The less noisy steps therefore need the pair loss as well. With the pair loss at every step, the model follows 59.4% of the template edits and 56.0% of the free-form edits. Without vocabulary swaps template-edit success is 58.5% and without rare-position oversampling, with all clips drawn equally often, 57.9%.

5. Conclusion

We introduced MinPair, which builds the supervision for controllable text-to-motion generation from measurements of real motion clips. The same rule-based motion descriptors yield MinPair captions and the measured minimal pairs of a pair loss. With the same network, MinPair captions improve attribute accuracy, M2M, R@3 and FID over FrankenMotion captions, yet the denoiser still follows only 11% of the template edits. The pair loss raises this share to 60% while the three standard metrics do not improve further. Controlling a single attribute is thus a separate skill from fitting a complete caption. A model acquires it reliably only when one word alone separates two otherwise identical captions. With body-part captions MinPair also reaches more targets than state-of-the-art methods trained on far more motion, so targeted supervision can outweigh scale.

Impact. MinPair turns the words of a caption into controls. An animator can change how high, how fast or how far a movement goes by editing one word. The same control applies when a robot imitates the generated motion and its operator states the execution in words. The supervision needs neither new recordings nor new human annotation, only computation, so it scales with the size of the corpus. The pair loss adds one term to training and leaves inference unchanged, so existing diffusion generators can adopt it. The two benchmarks measure the control that standard metrics miss and apply to any text-to-motion model. Measured minimal pairs are not tied to motion. They can be built wherever the property behind a word can be measured.

Limitations. MinPair covers attributes that can be measured from joint positions and expressed by a word scale or a short phrase. Style, emotion and contact with objects lie outside this set. We trained the pair loss with one part-track denoiser on one corpus, so other generator families remain to be tested. Furthermore, applying the pair loss to all clips of the batch remains unexplored.

Future work. So far the measured words act through the part tracks. The next step carries them into the global caption so that a single sentence steers the execution as well. Further pairs can change when a movement happens or how several body parts move together, as long as each pair stays checkable against the measurements. The creation of the supervision can also become agentic, with language-model agents that find what the generator still misses and add the measurements and pairs for it.

Acknowledgments

This research was funded by the European Regional Development Fund (ERDF) within the project 'AIDA' (grant FEIH 2476184). Views and opinions expressed are those of the authors only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Writing and coding assistants (Claude by Anthropic, ChatGPT by OpenAI) helped to draft and revise this manuscript, write and run experiment code and prepare tables and figures. The authors verified every number, claim and reference and take full responsibility for the content.

References

[1] Nikos Athanasiou, Alpár Cseke, Markos Diomataris, Michael J. Black, and Gül Varol. MotionFix: Text-driven 3D human motion editing. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024. 3

[2] Bin Cao, Sipeng Zheng, Hao Luo, Boyuan Li, Jing Liu, and Zongqing Lu. OpenT2M: No-frill motion generation with open-source, large-scale, high-quality data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 30640–30649, 2026. 3

[3] Wenshuo Chen, Hongru Xiao, Erhang Zhang, Lijie Hu, Lei Wang, Mengyuan Liu, and Chen Chen. SATO: Stable text-to-motion framework. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 6989–6997, 2024. 3

[4] Ginger Delmas, Philippe Weinzaepfel, Thomas Lucas, Francesc Moreno-Noguer, and Grégory Rogez. PoseScript: 3D human poses from natural language. In *ECCV*, 2022. 2, 3

[5] Ginger Delmas, Philippe Weinzaepfel, Francesc Moreno-Noguer, and Grégory Rogez. PoseFix: Correcting 3D human poses with natural language. In *ICCV*, pages 15018–15028, 2023. 3

[6] Jeonghyeok Do and Munchurl Kim. Bridging the skeleton-text modality gap: Diffusion-powered modality alignment for zero-shot skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12757–12768, 2025. 3

[7] Kent Fujiwara, Mikihiro Tanaka, and Qing Yu. Chronologically accurate retrieval for temporal grounding of motion-language models. In *ECCV*, pages 323–339, 2024. 3

[8] 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*, pages 5152–5161, 2022. 1, 2, 3, 4, 5

[9] Ke Han, Yueming Lyu, and Nicu Sebe. Motion-aligned word embeddings for text-to-motion generation. In *International Conference on Learning Representations*, pages 28276–28289, 2026. 3

[10] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. arXiv:2207.12598, 2022. 5

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, pages 6840–6851, 2020. 2, 4

[12] Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. SugarCrepe: Fixing hackable benchmarks for vision-language compositionality. In *NeurIPS*, pages 31096–31116, 2023. 3

[13] Yiming Huang, Weilin Wan, Yue Yang, Chris Callison-Burch, Mark Yatskar, and Lingjie Liu. CoMo: Controllable motion generation through language guided pose code editing. In *ECCV*, pages 180–196, 2024. 3

[14] Nan Jiang, Yunhao Li, Lexi Pang, Zimo He, Siyuan Huang, and Yixin Zhu. MotionMaster: Generalizable text-driven motion generation and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 30629–30639, 2026. 3

[15] Amita Kamath, Cheng-Yu Hsieh, Kai-Wei Chang, and Ranjay Krishna. The hard positive truth about vision-language compositionality. In *ECCV*, pages 37–54, 2024. 5

[16] Heechang Kim, Gwanghyun Kim, and Se Young Chun. LaMoGen: Laban movement-guided diffusion for text-to-motion generation. arXiv:2509.24469, 2025. 3

[17] Benno Krojer, Elinor Poole-Dayan, Vikram Voleti, Christopher Pal, and Siva Reddy. Are diffusion models vision-and-language reasoners? In *Advances in Neural Information Processing Systems*, pages 8385–8405, 2023. 3, 5

[18] Rudolf Laban and Lisa Ullmann. *The Mastery of Movement*. Macdonald & Evans, London, 3rd edition, 1971. 3

[19] Jaa-Yeon Lee, Byunghee Cha, Jeongsol Kim, and Jong Chul Ye. Aligning text to image in diffusion models is easier than you think. In *NeurIPS*, 2025. 3, 5

[20] Alexander C. Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2206–2217, 2023. 3

[21] Chuqiao Li, Xianghui Xie, Yong Cao, Andreas Geiger, and Gerard Pons-Moll. FrankenMotion: Part-level human motion generation and composition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16420–16431, 2026. 1, 2, 3, 4, 5, 7

[22] Xinpeng Liu, Yong-Lu Li, Ailing Zeng, Zizheng Zhou, Yang You, and Cewu Lu. Bridging the gap between human motion and action semantics via kinematic phrases. In *ECCV*, pages 223–240, 2024. 2, 3

[23] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *ICCV*, pages 5442–5451, 2019. 5

[24] 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 *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 27859–27871, 2025. 1, 3, 5, 7

[25] Massimiliano Pappa, Luca Collorone, Giovanni Ficarra, Indro Spinelli, and Fabio Galasso. MoDiPO: Text-to-motion alignment via AI-feedback-driven direct preference optimization. *Frontiers in Computer Science*, 8:1707808, 2026. 3

[26] Mathis Petrovich, Or Litany, Umar Iqbal, Michael J. Black, Gül Varol, Xue Bin Peng, and Davis Rempe. Multi-track timeline control for text-driven 3D human motion generation. In *CVPR Workshops*, pages 1911–1921, 2024. 3

[27] Abhinanda R. Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J. Black. BABEL: Bodies, action and behavior with English labels. In *CVPR*, pages 722–731, 2021. 2, 4

[28] Qwen Team. Qwen3.6-27B. https://huggingface.co/Qwen/Qwen3.6-27B, 2026. Model card, accessed 22 October 2026. 4

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021. 4

[30] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. 4

[31] 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:2603.15546, 2026. 1, 2, 3, 5, 7

[32] Keming Shen, Bizhu Wu, Junliang Chen, Xiaoqin Wang, and Linlin Shen. FineXtrol: Controllable motion generation via fine-grained text. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(11):8905–8913, 2026. 3

[33] Zhenwu Shi, Jingyu Gong, Peiwei Wang, Xingzan Wang, Tianwen Qian, Wenxi Li, Yuan Fang, Jiao Xie, Lizhuang Ma, and Shaohui Lin. Omni-supervised motion editing: Balancing change and invariance through positive-negative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 30595–30606, 2026. 3

[34] Xiaofeng Tan, Wanjiang Weng, Hongsong Wang, Fang Zhao, Xin Geng, and Liang Wang. MotionRFT: Unified reinforcement fine-tuning for text-to-motion generation. arXiv:2603.27185, 2026. 3

[35] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H. Bermano. Human motion diffusion model. In *ICLR*, 2023. 2, 3, 4

[36] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv:1807.03748, 2018. 3, 5

[37] 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. 1, 2, 3, 5, 7

[38] Bizhu Wu, Jinheng Xie, Meidan Ding, Zhe Kong, Jianfeng Ren, Ruibin Bai, Rong Qu, and Linlin Shen. FineMotion: A dataset and benchmark with both spatial and temporal annotation for fine-grained motion generation and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13837–13846, 2025. 3

[39] Bin Yang, Rong Ou, Weisheng Xu, Jiaqi Xiong, Xintao Li, Taowen Wang, Luyu Zhu, Yuetong Fang, Xu Jiang, Jing Tan, and Renjing Xu. T2MBench: Evaluating text-to-motion generation under diverse language instructions. In *Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1622–1662. Association for Computational Linguistics, 2026. 3

[40] Payam Jome Yazdian, Rachel Lagasse, Hamid Mohammadi, Eric Liu, Li Cheng, and Angelica Lim. MotionScript: Natural language descriptions for expressive 3D human motions. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 21574–21581, 2025. 2, 3

[41] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *ICLR*, 2023. 2, 3

[42] Mingyuan Zhang, Huirong Li, Zhongang Cai, Jiawei Ren, Lei Yang, and Ziwei Liu. FineMoGen: Fine-grained spatio-temporal motion generation and editing. In *Advances in Neural Information Processing Systems*, pages 13981–13992, 2023. 3

[43] Pengfei Zhang, Pinxin Liu, Pablo Garrido, Hyeongwoo Kim, and Bindita Chaudhuri. KinMo: Kinematic-aware human motion understanding and generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11187–11197, 2025. 3