CONTROLLABLE EXAGGERATION FOR GENERATIVE MOTION MODELS VIA TRAINING-TIME ADAPTATION AND INFERENCE-TIME GUIDANCE

Anonymous authors
Paper under double-blind review

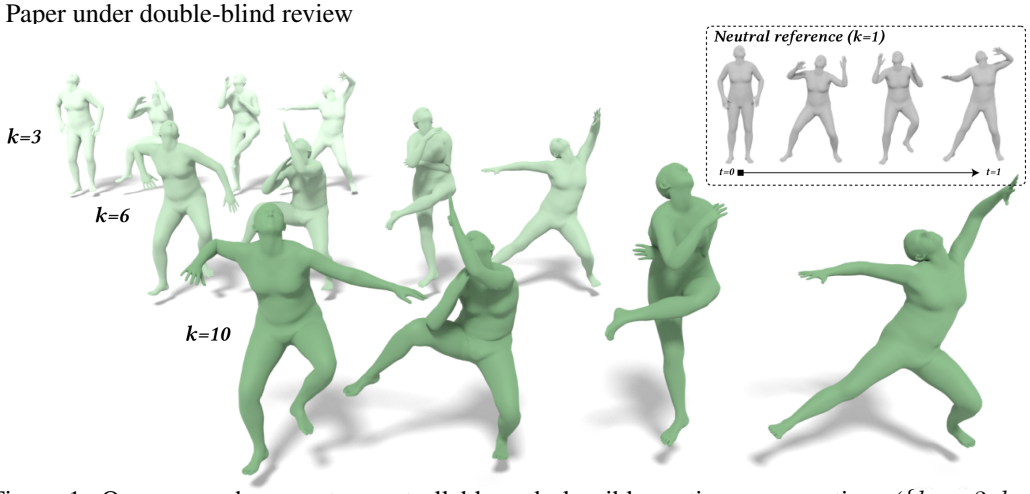


Figure 1: Our approach generates controllable and plausible motion exaggerations ($\{k = 3, k = 6, k = 10\}$ in green) from a neutral reference motion (dashed-line box in gray). Increasing k progressively amplifies the dance motion while preserving its underlying action intent.

ABSTRACT

Recent motion generative models have demonstrated strong capabilities in synthesizing physically plausible character motion, but often overlook established animation principles used by professional animators to ground and design their animation work. Understanding and incorporating these principles into motion generative pipelines is essential for producing motions that serve not only physically grounded applications but also the needs of the character animation community. This enables the creation of characters that not only move in physically plausible ways but also feel alive, expressive, and engaging. To close this gap, we focus on the *Exaggeration* principle of animation and investigate how it can be incorporated into modern motion generative pipelines to produce more expressive character motions. To this end, we introduce a framework that operates at two stages of existing motion generative pipelines. The first stage introduces exaggeration during training, where we perform supervised fine-tuning of pre-trained text-to-motion models on our curated exaggeration dataset. The second stage operates at inference time, where we: (i) introduce a mathematical formulation of exaggeration based on dynamic movement primitives (DMPs); and (ii) leverage this formulation as an exaggeration guidance signal to guide existing diffusion and flow-matching text-to-motion generation models toward exaggerated motion without additional training. Through qualitative and quantitative evaluations against three strong motion generation models, we show that our methods generate more exaggerated and expressive motions while preserving neutral reference motion intent and physical plausibility.

1 INTRODUCTION

Character animation is a long-standing field in which artists and animators design and combine different techniques to create character motion that conveys dynamic behavior and interactions with the environment, as seen in cartoons, films, and games. As a common practice, animators rely on a set of well-established principles, commonly known as the 12 Principles of Animation, which were proposed by Disney animators in their book *The Illusion of Life* (Johnston & Thomas, 1981).

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.

These principles provide a framework that guides animators in designing and refining character motion. Understanding and incorporating these principles into modern motion generation pipelines is essential for producing motions that are relevant not only to physically grounded applications but also to the character animation community (Lasseter, 1998). This integration could unlock more compelling generative character animation workflows, combining physically plausible motion with the expressiveness needed to make characters feel alive on screen. While recent motion generative foundation models (Tevet et al., 2023; Rempe et al., 2026; Wen et al., 2025; Tevet et al., 2025; Gat et al., 2025) produce realistic character motions, they often neglect the animation principles in the generation pipeline. Specifically, their focus is mainly on generating physically plausible motions that mimic real-world motion and match how humans and animals actually move. To close this gap, we target the *Exaggeration* principle of animation, which is one of the most relevant principles for presenting motion in an expressive way (Wu et al., 2026; Lasseter, 1998; Johnston & Thomas, 1981), and aim to generate motions that are exaggerated and expressive. We provide a description of the 12 animation principles in Appendix A. Incorporating the *Exaggeration* principle into a motion generative pipeline requires understanding how animators interpret exaggeration and how they apply it in practice when designing expressive motions for characters. To understand such requirements, we had discussions with professional animators, which revealed several important insights that we account for in our pipeline design. As a result of these discussions, we identify three properties that an exaggerated motion generation method should satisfy: (i) *controllable exaggeration*, where the animator should be able to adjust the motion exaggeration intensity through parameters that produce progressively more exaggerated motion; (ii) *motion-intent preservation*, where exaggeration should amplify the expressive shape of the motion without changing the underlying action. For example, an exaggerated punch should still read as the same neutral punch action, only with stronger movement quality; and (iii) *physical plausibility*, where increasing exaggeration should not introduce severe motion artifacts, such as invalid body poses, foot penetration, or unrealistic global movement. To address these requirements, we introduce a two-stage pipeline that incorporates exaggeration at the training and inference stages of existing motion generative foundation models. In the first stage, our *training-time exaggeration* approach teaches pre-trained motion generative models to directly learn the characteristics of exaggerated motion from real motion capture performances with different exaggeration intensities through supervised fine-tuning of the pre-trained models. We study this approach across different motion generative architectures, including diffusion models (Rempe et al., 2026; Tevet et al., 2023) and flow-matching models (Wen et al., 2025), using an exaggeration dataset that we design and construct specifically for this task. In the second stage, our *inference-time exaggeration* approach introduces exaggeration without modifying the parameters of the pre-trained model. Specifically, we mathematically formulate the *Exaggeration* principle of animation using dynamic movement primitives (DMPs) (Ijspeert et al., 2013) and use this formulation to steer the generation process of text-to-motion models toward progressively exaggerated motions.

The main contributions of this study are: *(i)* To the best of our knowledge, this is the first work to formulate and incorporate an animation principle into modern motion generative foundation models (Sec. 3); *(ii)* We present two complementary approaches for synthesizing exaggerated and expressive character motions, operating at training time and inference time of motion generative pipelines, respectively (Sec. 3.2 and Sec. 3.3); and *(iii)* We develop an evaluation suite that includes both classifier-based and classifier-free approaches for evaluating exaggerated motions (Sec. 4.2).

2 Related Work

Expressive Motion Generation. Existing approaches to expressive motion generation broadly focus on either characterizing expressiveness or manipulating motion style. A line of work seeks to quantify expressiveness so that it can be measured and controlled. Laban Movement Analysis (Samadani et al., 2013) is the foundation of this line of work: it maps raw kinematics to interpretable descriptors of effort and shape (Larboulette & Gibet, 2015; Samadani et al., 2013; Aristidou et al., 2015; Camurri et al., 2003), providing per-joint and per-frame measures of quantity of motion, weight, time, and spatial extent. Another line of work focuses on motion style editing, including unpaired motion style transfer (Aberman et al., 2020), body-part-level style control (Jang et al., 2022), and, most closely related to our work, procedural squash-and-stretch and smear effects (Basset et al., 2024). These methods either transfer a reference style or apply hand-authored effects, but they do not specifically model any animation principles such as exaggeration that we target in this work.

Motion Generative Models. Research on motion generative models has grown rapidly in recent years Khani et al. (2025); Zhu et al. (2023). Text- and kinematics-conditioned generative models

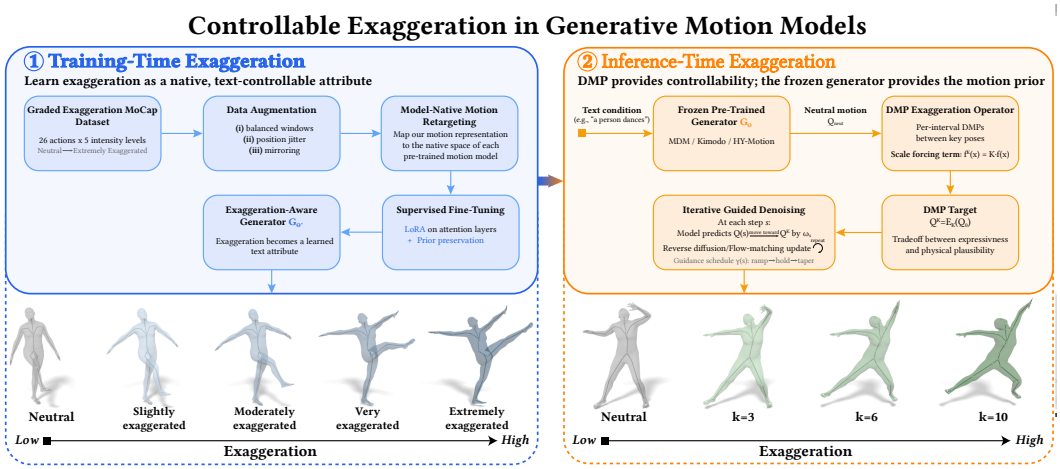


Figure 2: **An overview of the controllable motion exaggeration framework,** consisting of two stages: *(1) Training-Time Exaggeration*, constructs a graded motion-capture dataset with 26 actions and five exaggeration levels, augment and retarget the data, and perform supervised fine-tuning with LoRA and prior preservation to learn exaggeration as a controllable, text-conditioned attribute (Sec. 3.2); *(2) Inference-Time Exaggeration*, where a frozen pre-trained motion generation model first produces a neutral motion Q_{neut}, which is transformed by a DMP-based exaggeration operator $\mathcal{E}_K$ into an exaggerated target Q_K. Then, iterative guidance steers the sampling process toward Q_K, while the frozen generator regularizes the motion through its learned physical prior (Sec. 3.3). The bottom examples show progressively increasing levels of exaggeration.

produce whole-body motion from natural-language prompts (Rempe et al., 2026; Wen et al., 2025; Tevet et al., 2023; Gat et al., 2025). They offer control through text and an extensive set of kinematic constraints, including full-body pose keyframes, end-effector targets, and 2D paths, while operating on carefully designed motion representations. While the motions synthesized by these generative models are mostly physically plausible and realistic, they may not be suitable for character animation applications, where characters should feel alive and engaging. In this work, we leverage the neutral motions synthesized by these motion foundation models as reference motions and aim to make them more expressive and exaggerated.

Dynamic Movement Primitives. DMPs (Ijspeert et al., 2013; Saveriano et al., 2023) model a movement as a stable second-order dynamical system. DMPs have been widely used in robotics for imitation learning (Ijspeert et al., 2013; Pastor et al., 2013), generalization (Ginesi et al., 2021), and, most relevant to our work, expressive robot motion (Hielscher et al., 2025). In the context of character and robot motion, where rotations live on a non-Euclidean manifold, DMPs have also been formulated for quaternions (Ude et al., 2014; Koutras & Doulgeri, 2020). In our work, we fit DMPs either on Euler channels, on the continuous 6D rotation representation (Zhou et al., 2019), or directly on the motion generator's native representation. To our knowledge, modeling the *Exaggeration* principle of animation using DMPs has been explored only for single start-goal robot trajectories in (Hielscher et al., 2025), but not for multi-keyframe 3D character animation. In this work, we leverage DMPs and motion generative models to close this gap.

3 Methodology

We introduce a framework for incorporating the *Exaggeration* principle of animation into generative motion models at two stages of the generative pipeline. First, we introduce exaggeration during training through supervised fine-tuning of pre-trained text-to-motion models on an exaggeration dataset that we construct, containing 26 actions, with five exaggeration levels per action, enabling the models to learn exaggeration as a controllable text property. Second, at inference time, we steer pre-trained motion generation models toward exaggerated motions without modifying their parameters. This stage consists of two steps: *(i)* formulating exaggeration mathematically using dynamic movement primitives (DMPs) (Ijspeert et al., 2013), and *(ii)* using this formulation as a guidance signal for diffusion (Rempe et al., 2026; Tevet et al., 2023) and flow-matching (Wen et al., 2025) text-to-motion models. An overview of the training-time and inference-time procedures is shown in Fig. 2 and Algorithm 1.

3.1 Problem Statement and Preliminaries

Let a character motion be represented as a sequence $Q = q_1, \ldots, q_T$, where q_t denotes the character skeleton representation at frame t.

We consider a conditional motion generative model $\mathcal{G}_\theta$ with parameters θ and an optional semantic condition c, such as a text prompt. A neutral motion generated by the pre-trained model is $Q_{\text{neut}} = \mathcal{G}_\theta(c)$. Given an exaggeration intensity K, our goal is to generate Q_K whose expressive characteristics increase with K while preserving the semantic content and physical properties of Q_{neut}. Specifically, exaggeration should provide: (i) *controllable exaggeration*, where $K = 1$ recovers the neutral motion and $K > 1$ progressively increases exaggeration; (ii) *motion-intent preservation*, where exaggeration changes *how* an action is performed without changing *what* action is performed; and (iii) *physical plausibility*, where increasing exaggeration does not introduce severe artifacts such as invalid body poses, foot penetration, or unrealistic global movement.

Algorithm 1 Controllable Motion Exaggeration

Stage 1: Training-Time Exaggeration (Sec. 3.2)
Retarget the ExMoCap dataset $\mathcal{D}_{\text{exag}} = \{(Q^{(\ell)}, c_\ell)\}_{\ell=0}^{4}$
Apply motion augmentation
Initialize $\theta \leftarrow \theta_0$ and insert LoRA adapters of rank r
for $n = 1$ **to** N **do**
  Sample $(Q^{(\ell)}, c_\ell)$, diffusion step s, and noise ϵ
  Compute the diffusion training loss $\mathcal{L}_{\text{SFT}}$ using $\mathcal{G}_\theta$
  Compute the prior-preservation loss $\mathcal{L}_{\text{prior}}$
  $\mathcal{L} \leftarrow \mathcal{L}_{\text{SFT}} + \lambda\, \mathcal{L}_{\text{prior}}$
  Update the trainable LoRA parameters: $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$
end for
$\theta^\star \leftarrow \theta$ ▷ fine-tuned exaggeration model

Stage 2: Inference-Time Exaggeration (Sec. 3.3)
Set $\theta_{\text{inf}} \leftarrow \theta_0$ for TF, or $\theta_{\text{inf}} \leftarrow \theta^\star$ for SFT+TF
$z_S \sim \mathcal{N}(0, I)$; sample neutral motion Q_{neut}
$Q_K \leftarrow \mathcal{E}_K(Q_{\text{neut}})$ ▷ DMP-exaggerated target
Reset z_S using the same seed as Q_{neut}
for $s = S$ **to** 1 **do**
  $\widehat{Q}^{(s)} \leftarrow \mathcal{G}_{\theta_{\text{inf}}}(z_s, s, c)$ ▷ clean-motion prediction
  $\omega_s \leftarrow \omega\, \gamma(s)$
  $Q_e^{(s)} \leftarrow \widehat{Q}^{(s)} + \omega_s \left(Q_K - \widehat{Q}^{(s)}\right)$
  $Q_e^{(s),\text{root}} \leftarrow Q_0^{\text{root}}$ ▷ root lock
  $z_{s-1} \leftarrow \text{DDIM}(z_s, Q_e^{(s)})$
end for
return $Q_e^{(0)}$

3.2 Training-Time Exaggeration

Dataset Construction. One way to model exaggeration is to learn how humans actually perform exaggerated motions. To this end, we recruited two professional actors to perform different exaggerated and expressive movements and constructed a motion-capture dataset containing 26 actions, with each action performed at five different levels of exaggeration. We refer to this dataset as **ExMoCap**. We denote the exaggeration level by $\ell \in \{0, 1, 2, 3, 4\}$, where $\ell = 0$ corresponds to the neutral version of the action and $\ell = 4$ corresponds to its most exaggerated version. This graded structure provides supervision for learning how the movement quality of an action changes as exaggeration increases while its motion intent remains unchanged.

SFT Formulation. Then, we perform supervised fine-tuning of pre-trained motion generation models using ExMoCap, where exaggeration intensity is introduced through the text conditioning. Specifically, each discrete level ℓ is associated with an intensity description ranging from neutral to extremely exaggerated and appended to the underlying action caption (e.g., '*a person dances, very exaggerated*"). This allows exaggeration to be learned as a text-conditioned attribute and enables users to control exaggeration at inference time by changing the intensity description while keeping the action unchanged. We fine-tune three pre-trained motion foundation models: MDM (Tevet et al., 2023), HY-Motion (Wen et al., 2025), and KIMODO (Rempe et al., 2026). Since these models use different motion representations, we first retarget each motion sample in ExMoCap to a common SMPL-22 body skeleton represented by 3D joint trajectories (Loper et al., 2015). Then, we resample and encode each trajectory separately in the native motion representation of each backbone, allowing every model to be fine-tuned in the same feature space and normalization used during its pre-training. Appendix D.3 and Table 3 summarize these representations. Let $Q^{(\ell)} \in \mathbb{R}^{T \times D}$ denote a motion at exaggeration level ℓ, with T frames and $D = 263$ for MDM's HumanML3D motion representation, and let c_ℓ denote its corresponding action-exaggeration text condition (e.g., '*a person dances, very exaggerated*"). During fine-tuning of MDM (Tevet et al., 2023), we sample a diffusion timestep s and Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, and construct the corresponding noisy motion z_s as:

$$z_s = \sqrt{\bar{\alpha}_s}\, Q^{(\ell)} + \sqrt{1 - \bar{\alpha}_s}\, \epsilon, \qquad \bar{\alpha}_s = \prod_{u=1}^{s} (1 - \beta_u), \tag{1}$$

where $\{\beta_u\}$ is the diffusion noise schedule inherited from the pre-trained checkpoint. Then, in the reverse diffusion process, the model $\mathcal{G}_\theta(z_s, s, c_\ell)$ predicts the clean motion $\hat{Q}^{(\ell)}$ from the noisy intermediate variable z_s: $\hat{Q}^{(\ell)} = \mathcal{G}_\theta(z_s, s, c_\ell)$. We fine-tune the model using the clean-motion reconstruction objective of MDM (Tevet et al., 2023):

$$\mathcal{L}_{\text{SFT}} = \mathbb{E}_{Q^{(\ell)}, s, \epsilon} \left[\frac{1}{D \sum_t M_t} \sum_{t=1}^{T} \sum_{d=1}^{D} M_t \left(Q^{(\ell)} - \mathcal{G}_\theta(z_s, s, c_\ell) \right)_{t,d}^2 \right], \tag{2}$$

where M is a validity mask excluding padded frames. The formulation for supervised fine-tuning of HY-Motion and KIMODO, which use flow-matching and diffusion-based generative formulations, respectively, is provided in Appendices D.1 and D.2. We refer to this approach as *supervised fine-tuning (SFT)* throughout the paper.

LoRA and Prior-Preservation For Feasible Fine-Tuning. Since our exaggeration dataset is considerably smaller than the datasets used to pre-train the motion foundation models, naively fine-tuning all model parameters θ can lead to overfitting and catastrophic forgetting (Kirkpatrick et al., 2017; Hur et al., 2024), causing the model to lose its ability to generate diverse motions during fine-tuning (Table 6). To address this issue, inspired by (Hu et al., 2021; Ruiz et al., 2023; Sawdayee et al., 2026), we adopt three techniques. First, *data augmentation*: we augment the captured motions using left-right mirroring, balanced windowing, and position jittering, significantly increasing the number of samples in ExMoCap. Details of data augmentation techniques are provided in Appendix D.4. Second, *low-rank adaptation (LoRA)*: we freeze the original model weights and learn only low-rank updates to selected linear layers of each backbone (Hu et al., 2021), restricting the adaptation task to a smaller parameter space and helping retain the original motion prior. Third, *prior preservation* (Ruiz et al., 2023): fine-tuning batches additionally include motions from the original pre-training dataset, using the same denoising objective as Eq. 2.

3.3 Inference-Time Exaggeration

Another way to generate expressive motions is to formulate and incorporate the *Exaggeration* principle of animation into a pre-trained generative model through its inference process. The idea is to manipulate the diffusion or flow-matching process of motion generative models toward generating more exaggerated motions. However, this raises a natural question: what is the target exaggerated motion toward which the inference sampling process should be guided? To answer this question, in the next subsections, we propose an exaggeration operator $\mathcal{E}_K$ to formulate exaggeration and leverage this formulation as a guidance signal inside the generative sampling process during inference.

3.3.1 Mathematical Formulation of Exaggeration

We propose to formulate the exaggeration as geometric transformation of a neutral motion. Specifically, given a neutral motion Q_{neut}, we define an exaggeration operator, $Q_K = \mathcal{E}_K(Q_{\text{neut}})$, where K is a scalar controlling the exaggeration. We design $\mathcal{E}_K$ such that $K = 1$ reconstructs the neutral motion and $K > 1$ progressively exaggerates its movement characteristics while preserving the underlying action. To this end, we leverage dynamic movement primitives (DMP) Ijspeert et al. (2013), presented below.

Dynamic Movement Primitives Formulation. DMPs represents a movement as a stable dynamical system that combines *(i)* a goal-directed spring-damper term, which drives the motion from a start state to a goal state and thereby guarantees that the goal is reached, and *(ii)* a learned forcing term, $f(x)$, which captures the shape, swings, and arcs the motion makes on its way to the goal. In this work, we use the DMP formulation of Ijspeert et al. (2013), in which, for a single motion channel y, the DMP transformation system is given by the following differential equation:

$$\tau^2 \ddot{y} = \alpha\big(\beta(g - y) - \tau \dot{y}\big) + f(x), \tag{3}$$

where y_0 and g denote the start and goal states, α and β are gains that control the spring-damper dynamics (critically damped for $\beta = \alpha/4$), τ controls the temporal scale, and x is a phase variable that monotonically decays from 1 to 0, $\tau \dot{x} = -\alpha_x x$, $x(0) = 1$. The nonlinear forcing term $f(x)$ can be represented using a set of Gaussian basis functions $\psi_i(x)$,

$$f(x) = \frac{\sum_{i=1}^{N_w} \psi_i(x) w_i}{\sum_{i=1}^{N_w} \psi_i(x)}\, x(g - y_0), \tag{4}$$

where $\psi_i(x) = \exp(-h_i(x - c_i)^2)$ and w_i denotes the learnable weight of basis function i, which is obtained in closed form using locally weighted regression (Atkeson et al., 1997a;b) such that $f(x) \approx f_{\text{target}}(t)$. This formulation provides an interpretation for exaggeration. Specifically, manipulating the forcing term allows us to amplify the expressive shape of a trajectory.

Per-Interval DMP Motion In-Betweening. Conventional DMPs, which have been studied primarily in robotics (Ijspeert et al., 2013; Ginesi et al., 2021; Pastor et al., 2013; Hielscher et al., 2025; Saveriano et al., 2023), describe a single start-goal pair. However, a character motion violates this assumption because a motion sequence contains multiple key poses, i.e., multiple start-goal pairs, and naively fitting a single DMP to the entire motion sequence fails to accurately reconstruct the

motion. To address this issue, we fit one DMP independently to every keyframe interval $[t_i, t_{i+1}]$ in a motion sequence. For each interval and each motion channel, we set y_0 and g to the corresponding values of the two bounding poses. Given the neutral motion trajectory y_{neut} obtained from the pre-trained motion model, the target forcing term can be learned by rearranging Eq. 3,

$$f_{\text{target}}(t) = \tau^2 \ddot{y}_{\text{neut}}(t) - \alpha\big(\beta(g - y_{\text{neut}}(t)) - \tau \dot{y}_{\text{neut}}(t)\big). \tag{5}$$

Then, we estimate the basis weights w_i in Eq. 4 such that $f(x)$ reproduces f_{target}. Since one DMP is fitted per keyframe interval, naively resetting each DMP with zero velocity introduces a "stop-and-go" effect at interval boundaries. We avoid this by initializing each interval with the terminal velocity of the preceding interval. This maintains velocity continuity through the key poses.

Exaggerating Motion using DMP. Learning the forcing term $f(x)$ in Eq. 4, extracts the shape of motion for every joint in the character skeleton. To exaggerate this motion, inspired by Hielscher et al. (2025), one way is to scale the forcing term by a factor,

$$f^K(x) = K f(x). \tag{6}$$

Then, by replacing the forcing term in Eq. 3 with the scaled forcing term from Eq. 6, we obtain:

$$\tau^2 \ddot{y}_{\text{exag}} = \alpha\big(\beta(g - y_{\text{exag}}) - \tau \dot{y}_{\text{exag}}\big) + K f(x), \tag{7}$$

Then, given the scaled forcing term $K f(x)$, we solve Eq. 7 to obtain the updated exaggerated motion trajectory, including the joint positions y_{exag}, velocities $\dot{y}_{\text{exag}}$, and accelerations $\ddot{y}_{\text{exag}}$. Then, the exaggerated motion is the updated joint positions y_{exag} obtained from this process. The entire process can be summarized as:

$$Q_K = \mathcal{E}_K(Q_{\text{neut}}), \tag{8}$$

where Q_{neut} is the neutral motion, $\mathcal{E}_K$ represents all the steps from Eq. 3 to Eq. 7, and Q_K denotes the DMP-exaggerated motion. This gives K an interpretation as the exaggeration intensity in which $K = 1$ reconstructs the original neutral motion, while $K > 1$ amplifies the motion shape.

3.3.2 Exaggeration Formulation for Guiding Diffusion and Flow Matching

The DMP provides a controllable mechanism for exaggeration but has no knowledge of physically plausible character motion. Therefore, naively exaggerating using DMP (Eq. 8), especially at large K, can introduce artifacts such as invalid body poses and unrealistic movements. To address this issue, we leverage a pre-trained motion generation model to iteratively refine the DMP-exaggerated motion, keeping it close to the physically plausible motion manifold learned during pre-training. This is possible because motion foundation models learn a strong prior over realistic motion capture data and implicitly capture the physical properties of character motion during inference. However, they do not explicitly model the animation principle of exaggeration. Therefore, we combine DMPs with a pre-trained motion generation model to generate motions that are both expressive/exaggerated and physically plausible. In this way, the DMP determines the exaggerated trajectory (Eq. 8), while the pre-trained model iteratively regularizes it using its learned physical motion prior. We refer to this approach as *training-free (TF)*, or inference-time exaggeration, since it modifies only the inference process and requires no model training. We consider both diffusion (Rempe et al., 2026; Tevet et al., 2023) and flow-matching (Wen et al., 2025) motion generation models. The main paper focuses on the diffusion formulation. The flow-matching formulation is provided in Appendix C.

Integrating DMP-based exaggerated motion (Q_K) into diffusion denoising process. A motion diffusion model defines a forward process that gradually corrupts clean motion into Gaussian noise and a learned reverse process that reconstructs it. Following the clean-signal (x_0) parameterization of MDM (Tevet et al., 2023), we denote the noisy motion latent at diffusion step s by z_s and the model's clean-motion prediction by $\hat{Q}^{(s)} = \mathcal{G}_\theta(z_s, s, c)$. Sampling starts from $z_S \sim \mathcal{N}(0, I)$ and iteratively denoises the sample to obtain the final clean motion at $s = 0$. Our inference-time approach modifies this sampling process while keeping θ unchanged. Given a condition c, we first use the frozen model $\mathcal{G}_\theta$ to generate a neutral motion Q_{neut}. Then, we use Eq. 8 to obtain the DMP-based exaggerated motion Q_K. Then, we repeat this reverse diffusion process using the same initial noise that generated Q_{neut}. At each denoising step s, the frozen model predicts a cleaner motion, $\hat{Q}^{(s)} = \mathcal{G}_\theta(z_s, s, c)$, by removing noise from the output of the previous denoising step $s - 1$. To exaggerate this motion, before performing the next diffusion update, we move the prediction $\hat{Q}^{(s)}$ toward the DMP-exaggerated target Q_K. We repeat this process for S diffusion denoising steps. This process can be writtes as below:

$$\widetilde{Q}^{(s)} = \hat{Q}^{(s)} + \omega_s\big(Q_K - \hat{Q}^{(s)}\big), \qquad z_{s-1} = \text{DDIM}\big(z_s, \widetilde{Q}^{(s)}\big). \tag{9}$$

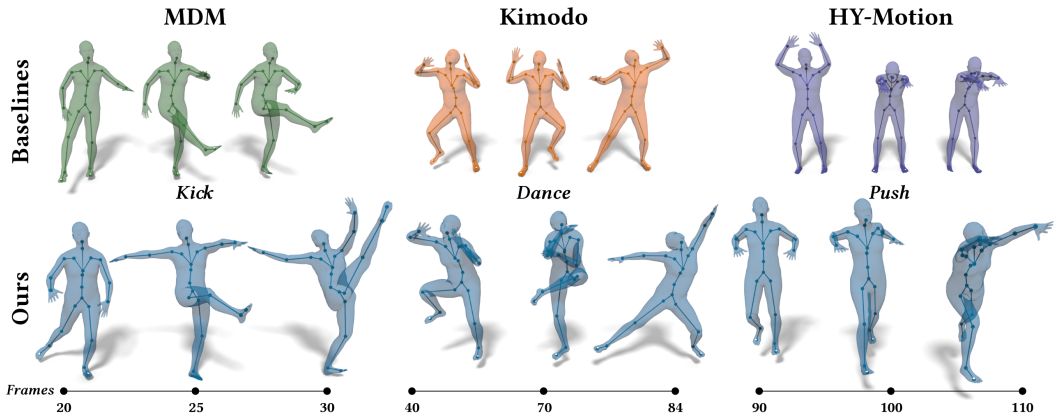


Figure 3: **Qualitative comparison of motion exaggeration.** For each action, the top row shows the motion generated by the pre-trained model, while the bottom row shows the corresponding motion generated with our exaggeration method. Across MDM (green), KIMODO (orange), and HY-Motion (purple), our method (blue) consistently produces more expressive and exaggerated movements while preserving the underlying action.

where ω_s denotes the exaggeration guidance strength at diffusion step s. We parameterize it as $\omega_s = \omega\,\gamma(s)$ where ω is the overall guidance scale and $\gamma(s) \in [0, 1]$ controls how strongly the exaggeration target is applied throughout the denoising trajectory. We use a heuristic schedule where the guidance remains weak during the earliest, highly noisy diffusion steps, and instead becomes stronger during the intermediate part of the generation process, and decreases again near the final denoising steps. This design was chosen to avoid forcing unreliable early predictions toward the target and gives the model additional freedom in the final steps to produce a clean and plausible motion. Eq. 9 balances two objectives: the difference $Q_K - \hat{Q}^{(s)}$ moves the current prediction toward the DMP-exaggerated motion, while passing the guided state through $\mathcal{G}_\theta$ at the next denoising step reintroduces the learned motion prior. In this way, the iterative refinement process corrects implausible poses introduced by DMP-based exaggeration (Eq. 8).

4 Experiments and Results

We evaluate our training-time (SFT), inference-time (TF), and combined (SFT+TF) exaggeration approaches across a diverse collection of character motions. We apply these approaches to three pre-trained text-to-motion backbones including KIMODO (Rempe et al., 2026), HY-Motion-1.0 (Wen et al., 2025), and MDM (Tevet et al., 2023). We evaluate the resulting motions through qualitative comparisons, quantitative metrics, ablation studies, and a perceptual user study. Additional experimental results are provided in Appendix E.

4.1 Qualitative Evaluation

Figs. 1 and 3 present representative qualitative comparisons for three diverse actions including kick, dance, and push across three baselines: MDM, KIMODO, and HY-Motion. Compared to the baselines, our approaches produce more expressive and exaggerated motions while preserving the underlying actions. Additional results covering more diverse actions, and exaggeration approaches (TF, SFT, and SFT+TF) are provided in Figs. 6–11 in Appendix E. We also encourage readers to view the supplemental video for additional qualitative comparisons with the baselines and for a more complete assessment of the resulting motions and their temporal progression.

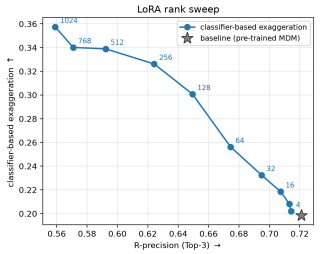
4.2 Quantitative Evaluation

We evaluate the proposed motion exaggeration methods along two evaluation axes: *(i) exaggeration ability*, where we assess how well the method can produce exaggerated and expressive motions, and *(ii) action preservation*, where we assess how well the method can preserve the intent and physical properties of the motion. An ideal expressive motion generation method should achieve strong exaggeration while preserving the underlying action and motion quality. However, stronger exaggeration naturally shifts the generated motions away from the original training distribution, which can affect distribution-based metrics such as FID even when the action remains recognizable and the motion physically plausible. Our results in Table 1 show that the proposed methods remain competitive on standard action-preservation and quality metrics while consistently achieving stronger exaggeration, with the combined SFT+TF approach providing the strongest overall exaggeration performance across the evaluated models.

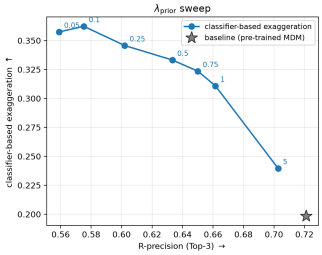
Table 1: **Quantitative evaluation of exaggeration and action preservation.** We report results for HY-Motion (1B), MDM, and KIMODO under the baseline, DMP, TF, SFT, and combined SFT+TF settings. The Exaggeration metrics measure exaggeration control and physical intensity, while Action Preservation metrics evaluate text-motion alignment and preservation of the learned motion manifold. Quality reports foot-skate artifacts. ↑ and ↓ indicate higher and lower values are better, respectively. Bold values highlight the best result within each model and metric, while underlined values indicate the second-best result.

		Exaggeration					Action Preservation				Quality
Model	**Method**	ELRA@1 ↑	ρ_{clf} ↑	ρ_{phys} ↑	Δ_{phys} ↑	Nov.↑	R@3↑	FID↓	Div↑	MMDist↓	Skate↓
MDM	Baseline	0.200	0.000	0.026	0.175	0.117	0.7031	**0.8388**	**9.5722**	**3.6133**	**0.070**
	+ DMP	0.200	0.000	0.358	0.953	0.130	0.5625	20.9116	6.8088	5.2530	0.134
	+ TF	0.200	0.000	0.252	0.552	0.023	**0.7070**	5.7547	8.3935	3.9870	0.091
	+ SFT	**0.408**	0.635	0.399	1.173	0.168	0.5547	1.6349	8.8127	4.6498	0.073
	+ SFT+TF	**0.408**	**0.643**	**0.545**	**1.610**	**0.276**	0.5273	5.8575	7.7830	5.0234	0.085
HY-Motion-1B	Baseline	0.354	0.354	0.391	0.902	0.135	**0.7422**	**2.0825**	**8.9325**	**3.4851**	**0.029**
	+ DMP	0.200	-0.042	0.516	1.318	0.167	0.4023	30.8515	4.7568	6.0145	0.061
	+ TF	0.200	0.007	0.424	1.017	0.082	0.5898	17.0509	6.4124	4.8573	0.057
	+ SFT	**0.938**	**0.985**	0.556	1.622	0.222	0.7383	2.2297	8.4439	3.6413	0.040
	+ SFT+TF	0.900	0.973	**0.697**	**2.003**	**0.301**	0.5430	19.7208	5.8073	5.1643	0.063
KIMODO	Baseline	0.223	-0.051	0.137	0.281	0.080	**0.6562**	2.0376	8.5198	**3.8190**	**0.025**
	+ DMP	0.208	0.053	0.374	1.086	0.144	0.3516	38.1156	5.7478	6.7370	0.069
	+ TF	0.208	0.039	0.287	0.801	0.081	0.4961	20.0688	6.8045	5.3506	0.036
	+ SFT	0.515	0.814	0.587	1.610	0.183	0.6289	**1.6934**	**8.9660**	3.9654	0.027
	+ SFT+TF	**0.531**	**0.831**	**0.729**	**2.015**	**0.306**	0.4648	18.9754	7.2581	5.5249	0.030

Motion Exaggeration Evaluation. To the best of our knowledge, this is the first work to incorporate the *Exaggeration* principle of animation into motion generation models and to introduce a dedicated evaluation suite for quantifying exaggeration in generated motions. Existing motion-generation benchmarks mainly evaluate text-motion alignment and generation quality, but do not assess whether a model produces the requested degree of exaggeration. To address this gap, we propose two complementary groups of metrics: *(i) classifier-based metrics*, which use an independent exaggeration-level classifier trained on ExMoCap to determine whether a generated motion exhibits the requested intensity level, and *(ii) classifier-free metrics*, which directly measure physical motion intensity from skeleton joint trajectories without relying on a learned classifier. The classifier-based group includes ELRA@k, $k \in \{1, 2, 3\}$, which measures how often the requested exaggeration level appears among the classifier's top-k predictions, and ρ_{clf}, the Spearman rank correlation (Spearman, 1987) between the requested and predicted levels, which measures whether the predictions follow the requested progression from neutral to extreme exaggeration. For the classifier-free metrics, we use root-relative range of motion (*ROM*) as the physical-intensity score, reducing the influence of global character translation. We report ρ_{phys}, the action-averaged Spearman rank correlation between the requested level and physical intensity, and Δ_{phys}, the standardized intensity difference between neutral and extremely exaggerated motions. These two metrics, ρ_{phys} and Δ_{phys}, capture the consistency and magnitude of physical amplification, respectively. Also, following (Wu et al., 2026), we report *Novelty* (*Nov.*), which measures the feature-space deviation between a highest-level exaggerated motion and its paired neutral reference using the pretrained HumanML3D motion encoder. Detailed definitions of all metrics are provided in Appendix B.



(a) Ablation on r



(b) Ablation on λ

Figure 4: **Ablation study on LoRA rank and prior-preservation weight.** We evaluate the trade-off between classifier-based exaggeration (ELRA@1) and prompt following by varying (a) the LoRA rank r and (b) prior-preservation weight λ for MDM. Higher ELRA@1 indicates stronger exaggeration control, while higher R-precision indicates better prompt adherence. The pre-trained MDM baseline is shown by the star.

Action-Preservation Evaluation. We evaluate action preservation and motion quality using the HumanML3D Guo et al. (2022) evaluation protocol. We report *R-precision@*k ($k \in 1, 2, 3$) and *MMDist* for text-motion alignment, *FID* and *Diversity* for motion-distribution quality, and *Foot-Skate* for physical plausibility. Lower values indicate better performance for *MMDist*, *FID*, and *Foot-Skate*, while higher values are better for the remaining metrics.

Ablation Studies. To analyze the design choices in our proposed method, we perform a set of controlled ablations and summarize the key findings below: *(i) Train-*

ing strategy. We compare two SFT strategies: full fine-tuning and the proposed LoRA+Prior-preservation fine-tuning recipe. Quantitative results in Tables 6-8 in Appendix E show that full fine-tuning produces more exaggerated motions but worse action preservation. In contrast, LoRA+Prior-preservation achieves a better trade-off, maintaining strong exaggeration while better preserving the action intent and type. *(ii) LoRA ranks and prior-preservation weights.* We ablate the proposed LoRA+Prior-preservation fine-tuning setting by training identical models with different values of LoRA rank r and prior-preservation weight λ. Fig. 4 shows that increasing the LoRA rank r improves exaggeration control (higher $\mathrm{ELRA@1}$) but reduces text adherence (lower R@3), while increasing the prior-preservation weight λ has the opposite effect. Therefore, we choose an operating point with $r \in \{64, 128\}$ and $\lambda \in \{0.1, 0.25\}$ depending on the base model to achieve a good trade-off. Tables 6-9 in Appendix E present the full ablation on r and λ for MDM, HY-Motion, and KIMODO. *(iii) Exaggeration components analysis.* We replace the exaggeration components proposed in this study. DMP, TF, SFT, and their composition (SFT+TF) are all designed to exaggerate character motion. As shown in Table 1, across all three motion models, DMP consistently increases physical exaggeration but substantially reduces action preservation and motion quality. Adding TF recovers action preservation and physical plausibility, demonstrating the value of the pre-trained motion prior for constraining DMP-exaggerated motion.

SFT provides stronger control over the requested exaggeration level while better preserving the original motion quality than DMP and SFT+TF. Finally, SFT+TF achieves the strongest physical exaggeration across the models, but at the cost of reduced action preservation and increased motion artifacts. Overall, the results demonstrate the expected exaggeration-preservation trade-off: DMP provides direct but unconstrained amplification, TF improves plausibility, SFT offers stronger exaggeration-level control, and SFT+TF maximizes physical exaggeration at the cost of greater deviation from the original action. The complete version of Table 1 is presented in Table 10 in Appendix E.

Table 2: **User-study results.** *Exaggeration* reports how often participants selected the highest requested level as the most exaggerated, while *Action Preservation* reports the percentage of positive preservation ratings.

Backbone	Exaggeration	Action Preservation
KIMODO	86.7%	82.0%
HY-Motion	78.9%	50.0%
MDM	70.0%	–
Overall	**81.8%**	**76.7%**

User Study. To perceptually assess the exaggeration capability of our method and its ability to preserve the underlying action, we conducted a user study with 20 participants with diverse backgrounds. For the *exaggeration* evaluation, participants viewed neutral and progressively exaggerated motions side-by-side and selected the most exaggerated and most neutral motions, resulting in 10 questions per participant. We report the *top-1 selection rate*, measuring how often the highest requested exaggeration level was selected as the most exaggerated, together with the neutral-reference selection rate. For the *action-preservation* evaluation, participants viewed a source motion and its exaggerated counterpart and judged whether the counterpart preserved the source action, resulting in 6 additional questions. Across the 16 evaluations (320 samples), the highest requested exaggeration level was selected as the most exaggerated in 81.2% of valid trials, while the exaggerated motions were judged to preserve the source action in 76.9% of cases. As summarized in Table 2, these results show that our approach produces clearly perceptible motion exaggeration while preserving the underlying action in most evaluations. The complete study protocol, representative questionnaire examples, and detailed results are provided in Appendix E, Fig. 12, and Table 5.

5 Conclusions

This study introduces a framework for incorporating the *Exaggeration* principle of animation into modern motion generative foundation models through training-time and inference-time approaches. Training-time exaggeration uses supervised fine-tuning on our graded exaggeration dataset, while inference-time exaggeration formulates exaggeration with dynamic movement primitives and uses the resulting signal to guide diffusion and flow-matching models without additional training. Qualitative and quantitative evaluations across multiple motion generation models demonstrate that our approaches generate more expressive motions while preserving the underlying action. Our work highlights the potential of incorporating established character animation principles beyond physical realism into motion generation. We hope this work encourages further research toward integrating additional animation principles into motion foundation models. One limitation of our current formulation is that the appropriate exaggeration scale is motion dependent. This means applying a very large K across diverse motions can occasionally produce jittery or atypical outputs. Future work could investigate reinforcement-learning-based post-training to improve motion plausibility while preserving the intended expressiveness.

REFERENCES

Kfir Aberman, Yijia Weng, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. Unpaired motion style transfer from video to animation. *ACM Transactions on Graphics (TOG)*, 39(4):64–1, 2020.

Andreas Aristidou, Panayiotis Charalambous, and Yiorgos Chrysanthou. Emotion analysis and classification: understanding the performers' emotions using the lma entities. In *Computer Graphics Forum*, volume 34, pp. 262–276. Wiley Online Library, 2015.

Christopher G Atkeson, Andrew W Moore, and Stefan Schaal. Locally weighted learning. *Artificial intelligence review*, 11(1):11–73, 1997a.

Christopher G Atkeson, Andrew W Moore, and Stefan Schaal. Locally weighted learning for control. *Artificial intelligence review*, 11(1):75–113, 1997b.

Jean Basset, Pierre Bénard, and Pascal Barla. Smear: Stylized motion exaggeration with art-direction. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*, 2024.

Antonio Camurri, Ingrid Lagerlöf, and Gualtiero Volpe. Recognizing emotion from dance movement: comparison of spectator recognition and automated techniques. *International journal of human-computer studies*, 59(1-2):213–225, 2003.

Inbar Gat, Sigal Raab, Guy Tevet, Yuval Reshef, Amit Haim Bermano, and Daniel Cohen-Or. Anytop: Character animation diffusion with any topology. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–10, 2025.

Michele Ginesi, Nicola Sansonetto, and Paolo Fiorini. Overcoming some drawbacks of dynamic movement primitives. *Robotics and Autonomous Systems*, 144:103844, 2021.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5152–5161, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Till Hielscher, Andreas Bulling, and Kai O Arras. Interactive expressive motion generation using dynamic movement primitives. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 18269–18276. IEEE, 2025.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pp. 492–518. Springer, 1992.

Jiwan Hur, Jaehyun Choi, Gyojin Han, Dong-Jae Lee, and Junmo Kim. Expanding expressiveness of diffusion models with limited data via self-distillation based fine-tuning. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5016–5025. IEEE, 2024.

Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2): 328–373, 2013.

Deok-Kyeong Jang, Soomin Park, and Sung-Hee Lee. Motion puzzle: arbitrary motion style transfer by body part. *ACM Transactions on Graphics (TOG)*, 41(3):1–16, 2022.

Ollie Johnston and Frank Thomas. The illusion of life: Disney animation. 1981.

Aliasghar Khani, Arianna Rampini, Bruno Roy, Larasika Nadela, Noa Kaplan, Evan Atherton, Derek Cheung, and Jacky Bibliowicz. Motion generation: A survey of generative approaches and benchmarks. *arXiv preprint arXiv:2507.05419*, 2025.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Leonidas Koutras and Zoe Doulgeri. A correct formulation for the orientation dynamic movement primitives for robot control in the cartesian space. *Conference on Robot Learning (CoRL)*, 2020.

Caroline Larboulette and Sylvie Gibet. A review of computable expressive descriptors of human motion. In *Proceedings of the 2nd International Workshop on Movement and Computing*, pp. 21–28, 2015.

John Lasseter. Principles of traditional animation applied to 3d computer animation. In *Seminal graphics: pioneering efforts that shaped the field*, pp. 263–272. 1998.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, October 2015.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Peter Pastor, Mrinal Kalakrishnan, Franziska Meier, Freek Stulp, Jonas Buchli, Evangelos Theodorou, and Stefan Schaal. From dynamic movement primitives to associative skill memories. *Robotics and Autonomous Systems*, 61(4):351–361, 2013.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

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

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

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22500–22510. IEEE, 2023.

Ali-Akbar Samadani, Sarahjane Burton, Rob Gorbet, and Dana Kulic. Laban effort and shape analysis of affective hand and arm movements. In *2013 Humaine Association conference on affective computing and intelligent interaction*, pp. 343–348. IEEE, 2013.

Matteo Saveriano, Fares J Abu-Dakka, Aljaž Kramberger, and Luka Peternel. Dynamic movement primitives in robotics: A tutorial survey. *The International Journal of Robotics Research*, 42(13): 1133–1184, 2023.

Haim Sawdayee, Chuan Guo, Guy Tevet, Bing Zhou, Jian Wang, and Amit H Bermano. Dance like a chicken: Low-rank stylization for human motion diffusion. In *Computer Graphics Forum*, pp. e70365. Wiley Online Library, 2026.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Charles Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 100(3/4):441–471, 1987.

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

Guy Tevet, Sigal Raab, Setareh Cohan, Daniele Reda, Zhengyi Luo, Xue Bin Peng, Amit Bermano, and Michiel Van de Panne. Closd: Closing the loop between simulation and diffusion for multi-task character control. In *International Conference on Learning Representations*, volume 2025, pp. 46506–46520, 2025.

Aleš Ude, Bojan Nemec, Tadej Petrić, and Jun Morimoto. Orientation in cartesian space dynamic movement primitives. *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2997–3004, 2014.

Yuxin Wen, Qing Shuai, Di Kang, Jing Li, Cheng Wen, Yue Qian, Ningxin Jiao, Changhai Chen, Weijie Chen, Yiran Wang, et al. Hy-motion 1.0: Scaling flow matching models for text-to-motion generation. *arXiv preprint arXiv:2512.23464*, 2025.

Leyi Wu, Pengjun Fang, Kai Sun, Yazhou Xing, Yinwei Wu, Songsong Wang, Ziqi Huang, Dan Zhou, Yingqing He, Ying-Cong Chen, et al. AnimationBench: Are video models good at character-centric animation? *arXiv preprint arXiv:2604.15299*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, pp. 5745–5753, 2019.

Wentao Zhu, Xiaoxuan Ma, Dongwoo Ro, Hai Ci, Jinlu Zhang, Jiaxin Shi, Feng Gao, Qi Tian, and Yizhou Wang. Human motion generation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4):2430–2449, 2023.

A REMARKS ON 12 PRINCIPLES OF ANIMATION

The 12 Principles of Animation were introduced by Disney animators as a set of principles for designing and producing character animation (Johnston & Thomas, 1981). These principles provide a practical framework for creating animation that is not only physically understandable, but also expressive and engaging to the audience. Although originally developed in the context of traditional hand-drawn animation, these principles remain relevant to computer animation and have been adapted to 3D character animation (Lasseter, 1998). Below, we briefly describe the 12 principles and their role in character animation.

1. Squash and Stretch describes the deformation of a character as it moves. Squashing and stretching can be used to communicate weight, flexibility, and the forces acting on a character. For example, a character may compress when landing and extend during a fast movement.

2. Anticipation prepares the audience for an upcoming action. A character typically performs a preparatory movement before executing the main action, such as bending the body before jumping or pulling an arm backward before throwing an object. Anticipation makes the subsequent action easier to perceive.

3. Staging concerns how an action, pose, or idea is presented so that its intention is clear to the audience. It involves the arrangement of the character, its pose, movement, and the surrounding scene to direct the viewer's attention toward the most important aspect of the action.

4. Straight Ahead Action and Pose-to-Pose describe two approaches to constructing an animated sequence. In straight-ahead animation, motion is created sequentially from one frame to the next, which can produce spontaneous and fluid motion. In pose-to-pose animation, important key poses are first established and the intermediate poses are subsequently generated between them.

5. Follow Through and Overlapping Action describe the fact that different parts of a character do not necessarily stop or change direction simultaneously. Parts such as the arms, legs, head, clothing, or other secondary elements may continue moving after the main body action has changed. This asynchronous movement helps produce more natural and dynamic character motion.

6. Slow In and Slow Out describes the gradual acceleration and deceleration of a movement. Rather than moving at a constant speed between two poses, a character often starts slowly, moves faster through the middle of the action, and slows down when approaching the next pose. This principle is commonly used to make motion feel smoother and more natural.

7. Arcs refers to the tendency of natural character movements to follow curved trajectories rather than perfectly straight paths. For example, the movement of an arm during a swing or the motion of a head during a turn generally follows an arc. Designing motion along appropriate arcs can make character movement appear more natural and fluid.

8. Secondary Action refers to an additional movement that supports and reinforces the main action without distracting from it. For example, while a character walks, arm movements, head motion, or changes in facial expression can provide additional information about the character's state and intention. Secondary actions can increase the expressiveness of a primary action.

9. Timing controls the temporal structure of an action, including how long a movement takes and how motion is distributed throughout the sequence. Different timing can communicate different properties, such as weight, speed, strength, emotion, or personality. Consequently, the same action can convey a different meaning or feeling when performed with different timing.

10. Exaggeration emphasizes or amplifies the characteristics of an action to make its intention, expression, or emotional quality clearer and more engaging. Importantly, exaggeration does not necessarily imply changing the underlying action. Instead, an animator can amplify the shape, timing, pose, or movement characteristics of an existing action while preserving its identity. For example, a punch can be exaggerated by increasing the extent or speed of its movement while still being recognized as the same punch action. The degree of exaggeration can range from subtle amplification to highly stylized motion, depending on the desired artistic effect.

11. Solid Drawing concerns the ability to construct and animate characters while maintaining a consistent understanding of their three-dimensional form, proportions, balance, weight, and perspective. Although originally formulated for hand-drawn animation, the principle emphasizes maintaining a coherent representation of the character as it moves through space.

12. Appeal refers to the visual and behavioral qualities that make a character interesting and engaging to the audience. An appealing character does not necessarily have to be attractive. Rather, its

design, expressions, and movements can also create a clear and compelling presence. In character animation, appeal can arise from both the character's appearance and the way the character moves.

B Remarks on the Exaggeration Evaluation Suite

To evaluate whether a generated motion exhibits the requested degree of exaggeration, we propose two complementary groups of metrics: *classifier-based* and *classifier-free*. Let N denote the number of evaluated motions, and let $l_i \in \{0, 1, 2, 3, 4\}$ denote the requested exaggeration level for motion i, ranging from neutral ($l_i = 0$) to extremely exaggerated ($l_i = 4$).

B.1 Classifier-Based Exaggeration Metrics.

Exaggeration-Level Classifier. We first evaluate whether a generated motion is recognized as having the requested exaggeration level. To this end, we train a motion classifier to predict the five exaggeration levels $l \in \{0, 1, 2, 3, 4\}$, corresponding to neutral, slightly, moderately, very, and extremely exaggerated motion. The classifiers are trained from scratch on real labeled motions from ExMoCap and do not receive action captions or generated motions as input. Once trained, their parameters remain frozen throughout evaluation. For MDM (Tevet et al., 2023), the classifier operates directly on the normalized 263-dimensional HumanML3D representation (Guo et al., 2022) used by the generative model. For HY-Motion (Wen et al., 2025) and KIMODO (Rempe et al., 2026), we instead use the 3D positions of the 22 body joints after subtracting the root position at every frame, resulting in a 66-dimensional translation-invariant representation. HY-Motion joint positions are obtained from the joint-position block of its native 201-dimensional representation, and KIMODO joint positions are decoded from its native 273-dimensional representation using its forward-kinematics decoder. Separate HY-Motion and KIMODO classifiers are trained because the motions are produced through different native representations and decoding procedures. The HY-Motion-0.4B and HY-Motion-1B models share the same classifier because they use the same motion representation.

All three classifiers use the same architecture. Given an input motion $\mathbf{X} \in \mathbb{R}^{C \times T_c}$, two one-dimensional convolutional layers with 256 channels, kernel size 3, and padding 1 are applied along the temporal dimension. Each convolution is followed by batch normalization and a ReLU activation. Then, adaptive temporal average pooling produces a single 256-dimensional motion descriptor, which is passed through dropout with probability 0.3 and a linear layer producing five level logits. We use $T_c = 100$ frames for MDM at 20 FPS and $T_c = 150$ frames for HY-Motion and KIMODO at 30 FPS, corresponding to a 5-second input window for every classifier. Longer training clips are randomly cropped. Each classifier is trained using five-class cross-entropy loss and the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 5×10^{-4}, weight decay 10^{-4}, and batch size 64. Training runs for at most 300 epochs, with early stopping after 40 epochs without improvement in exact validation accuracy.

During evaluation, every generated motion is converted to the corresponding classifier input representation, standardized using the classifier's stored statistics, and processed using the deterministic evaluation crop. Let $\mathbf{p}_i \in \mathbb{R}^5$ denote the resulting classifier logits for motion i, and let $\hat{l}_i = \arg\max_m p_{i,m}$ denote its predicted exaggeration level. The same frozen classifier checkpoint is used to evaluate every method and ablation associated with a given backbone. Because the classifiers operate in different representation spaces, their absolute scores are intended primarily for comparisons among methods evaluated on the same backbone.

Exaggeration-Level Recognition Accuracy. Using the frozen exaggeration-level classifier described above, we define ELRA@1 as

$$\mathrm{ELRA@1} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[l_i \in \mathrm{Top}_1(\mathbf{p}_i)], \tag{10}$$

where $\mathrm{Top}_1(\mathbf{p}_i)$ denotes the set of exaggeration-level indices corresponding to the classifier's largest logits for motion i, and $\mathbf{1}[\cdot]$ is the indicator function. Therefore, ELRA@1 measures the fraction of generated motions for which the requested exaggeration level is included among the classifier's top-1 predictions. Higher values indicate that generated motions more consistently exhibit the requested exaggeration level according to the frozen classifier.

Exaggeration-Level Rank Correlation. Because the exaggeration levels are ordinal, we additionally measure whether the classifier's predictions increase consistently as the requested level increases. We use Spearman's rank correlation coefficient (Spearman, 1987), which, for two sequences $\mathbf{u}$ and $\mathbf{v}$, is defined as

$$\rho_{\mathrm{S}}(\mathbf{u}, \mathbf{v}) = \mathrm{Corr}_{\mathrm{P}}(R(\mathbf{u}), R(\mathbf{v}))\,, \tag{11}$$

where $R(\cdot)$ converts the observations to ranks, assigning average ranks to tied values, and $\mathrm{Corr}_{\mathrm{P}}$ denotes Pearson correlation between the resulting ranks. Spearman correlation lies in $[-1, 1]$, where 1 indicates a perfectly monotonic increase, 0 indicates no monotonic association, and -1 indicates a perfectly monotonic decrease. Using the classifier's top-1 prediction $\hat{l}_i$, we define the classifier-based rank correlation as

$$\rho_{\mathrm{clf}} = \rho_{\mathrm{S}}\Big(\{l_i\}_{i=1}^{N}, \{\hat{l}_i\}_{i=1}^{N}\Big)\,. \tag{12}$$

While ELRA@1 measures per-motion agreement between the requested level and the classifier's predictions, ρ_{clf} measures whether the predicted levels follow the overall ordering of the requested exaggeration progression. Therefore, the two metrics are complementary: a model may preserve the correct ordering of exaggeration levels even when some individual levels are not classified exactly. A higher ρ_{clf} indicates more consistent control over the progression from neutral to extreme exaggeration.

B.2 Classifier-Free Exaggeration Metrics.

To complement the learned classifier-based metrics, we directly measure physical motion intensity from the generated skeleton joint trajectories.

Physical-Intensity Score. First, we use root-relative range of motion (*ROM*) as the primary physical-intensity measure, thereby reducing the influence of global character translation. Let $\mathbf{x}_{i,t,j} \in \mathbb{R}^3$ denote the position of joint j at frame t in motion i, and let r denote the root joint. We first compute the root-relative joint positions

$$\widetilde{\mathbf{x}}_{i,t,j} = \mathbf{x}_{i,t,j} - \mathbf{x}_{i,t,r}. \tag{13}$$

Then, the physical-intensity score for motion i is

$$s_i = \sum_{j=1}^{J} \left\| \max_t \widetilde{\mathbf{x}}_{i,t,j} - \min_t \widetilde{\mathbf{x}}_{i,t,j} \right\|_2, \tag{14}$$

where the maximum and minimum are computed independently along each spatial coordinate. This score measures the accumulated spatial range traversed by the joints relative to the character root. Larger values indicate motions with greater pose and limb displacement.

Let $\mathcal{A}$ denote the set of evaluated actions, and let $\mathcal{I}_a = \{i : a_i = a\}$ denote the set of samples corresponding to action a. We measure whether increasing the requested exaggeration level produces a monotonic increase in physical intensity using

$$\rho_{\mathrm{phys}} = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \rho_{\mathrm{S}}(\{l_i\}_{i \in \mathcal{I}_a}, \{s_i\}_{i \in \mathcal{I}_a})\,. \tag{15}$$

The correlation is computed separately within each action and then averaged, preventing actions with naturally larger ranges of motion from dominating the result. A higher ρ_{phys} indicates that increasing the requested exaggeration level more consistently produces greater physical motion intensity.

Physical-Intensity Separation. Correlation measures the ordering of the intensity levels but does not indicate the magnitude of the change between them. We therefore additionally measure the standardized separation between the neutral and extremely exaggerated levels. For each action a, we standardize its physical-intensity scores as

$$z_i = \frac{s_i - \mu_{a_i}}{\sigma_{a_i}}, \tag{16}$$

where μ_a and σ_a are respectively the mean and standard deviation of the physical-intensity scores across all evaluated samples and exaggeration levels for action a. Let

$$\bar{z}_\ell = \frac{1}{|\mathcal{I}_\ell|} \sum_{i \in \mathcal{I}_\ell} z_i, \qquad \mathcal{I}_\ell = \{i : l_i = \ell\}, \tag{17}$$

denote the mean standardized intensity at exaggeration level ℓ. We define the physical-intensity separation as

$$\Delta_{\text{phys}} = \bar{z}_4 - \bar{z}_0. \tag{18}$$

A larger Δ_{phys} indicates a greater physical separation between the neutral and extremely exaggerated motions. While ρ_{phys} measures whether intensity increases consistently across the full exaggeration range, Δ_{phys} measures the magnitude of the difference between its two endpoints.

Reference-Based Motion Novelty. Following the novelty principle introduced in AnimationBench (Wu et al., 2026), we measure how far an exaggerated motion deviates from its corresponding neutral reference in a learned motion-feature space. Let $\mathbf{f}_i^{\text{exag}}$ and $\mathbf{f}_i^{\text{neut}}$ denote the feature embeddings of an exaggerated motion and its paired neutral motion, respectively, extracted using the pretrained HumanML3D motion encoder. We compute

$$\text{Nov}_i = 1 - \frac{\langle \mathbf{f}_i^{\text{exag}}, \mathbf{f}_i^{\text{neut}} \rangle}{\|\mathbf{f}_i^{\text{exag}}\|_2 \, \|\mathbf{f}_i^{\text{neut}}\|_2}. \tag{19}$$

The paired motions use the same action condition and random seed, so that the score primarily reflects the change introduced by exaggeration rather than unrelated variation in generated content. We report the mean novelty of the highest requested exaggeration level relative to its corresponding neutral references. Larger values indicate a greater feature-space deviation from neutral motion. However, novelty measures the magnitude of this deviation and, by itself, does not determine whether the resulting motion is plausible or preserves the intended action.

Summary. The proposed metrics capture complementary aspects of controllable exaggeration. The classifier-based metrics, $\text{ELRA@}k$ and ρ_{clf}, measure whether generated motions are recognized as having the requested exaggeration levels. The classifier-free metrics, ρ_{phys} and Δ_{phys}, measure whether the requested progression is reflected in the ordering and magnitude of physical motion intensity. Finally, Nov. measures the feature-space deviation of the exaggerated motion from its neutral reference. Together, these metrics evaluate exaggeration control independently of conventional motion-quality and action-preservation metrics, which we report separately.

C Remarks on Training-Free Exaggeration Approach

Training-Free Exaggeration Formulation for Flow-Matching Models. HY-Motion (Wen et al., 2025) is a flow-matching motion model whose Diffusion Transformer (DiT) architecture (Peebles & Xie, 2023) predicts a velocity field that transports an initial Gaussian noise sample at flow time $t = 0$ to a clean motion at $t = 1$. Unlike MDM (Tevet et al., 2023) and KIMODO (Rempe et al., 2026), which directly predict the clean motion during reverse diffusion, HY-Motion predicts the instantaneous direction in which the current noisy state should evolve. Therefore, applying our training-free guidance to HY-Motion requires first converting its velocity prediction into an estimate of the clean motion.

Given a text condition c and initial noise $z_0 \sim \mathcal{N}(0, I)$, we first use the frozen HY-Motion model to generate a neutral motion Q_{neut}. Then, we apply the DMP-based exaggeration operator from Eq. 8 to obtain the exaggerated target $Q_K = \mathcal{E}_K(Q_{\text{neut}})$. Then, we repeat the flow-matching sampling process from the same initial noise z_0 used to generate Q_{neut}. Let z_t denote the current motion state at flow time t, and let $\mathcal{V}_\theta(z_t, t, c)$ denote the velocity predicted by the frozen HY-Motion model. Under the linear flow path

$$z_t = (1 - t)z_0 + tQ, \tag{20}$$

the corresponding estimate of the clean motion at $t = 1$ is

$$\widehat{Q}^{(t)} = z_t + (1 - t)\mathcal{V}_\theta(z_t, t, c). \tag{21}$$

This clean-motion estimate is the flow-matching analogue of the clean-motion prediction $\widehat{Q}^{(s)}$ used in the diffusion formulation (Sec. 3.3.2). Then, we guide this prediction toward the DMP-exaggerated target using

$$\widetilde{Q}^{(t)} = \widehat{Q}^{(t)} + \omega_t\big(Q_K - \widehat{Q}^{(t)}\big), \qquad \omega_t = \omega\,\gamma(t), \tag{22}$$

where ω is the overall guidance strength and $\gamma(t) \in [0, 1]$ is the same ramp-and-taper schedule used in the diffusion formulation (Sec. 3.3.2). As with MDM and KIMODO, the neutral root motion is restored after the guidance update so that exaggeration changes the local body motion without altering the character's global trajectory. Because HY-Motion advances the sample using a velocity rather than a clean-motion prediction, the guided clean estimate is converted back into a velocity before the next integration step. For $t < 1$, this gives

$$\widetilde{V}^{(t)} = \frac{\widetilde{Q}^{(t)} - z_t}{1 - t}, \qquad z_{t+\Delta t} = z_t + \Delta t\,\widetilde{V}^{(t)}. \tag{23}$$

Therefore, the DMP target modifies HY-Motion through its estimated clean endpoint. This means Eq. 22 moves the endpoint toward the desired exaggerated motion, and Eq. 23 translates that correction into the velocity used to advance the flow. In this way, subsequent evaluations of the frozen velocity model repeatedly reintroduce HY-Motion's learned motion prior. Consequently, the DMP provides the exaggeration direction while HY-Motion regularizes the motion throughout sampling, without requiring any update to the model parameters.

Exaggeration Control Knobs. The training-free exaggeration method has two control knobs: K determines the exaggeration level of Q_K relative to Q_{neut}, while ω controls how strongly the diffusion trajectory follows Q_K. Setting $K = 1$ recovers the neutral DMP trajectory, while $\omega = 0$ recovers the original sampling procedure.

Character's Root (Global) Trajectory Lock. Because our objective is mainly to exaggerate the local body motion rather than change where the character travels, the DMP does not amplify the character root's planar travel channels. However, during the guided diffusion process in the training-free approach, the model may alter character global movement to compensate for the increased local motion. To prevent such phenomena, we explicitly preserve the character root trajectory of the neutral sample by applying the original character's root trajectory to the motion being guided during the training-free approach:

$$\widetilde{Q}^{(s)}_{\text{root}} \leftarrow Q_{0,\text{root}} \tag{24}$$

after each guidance update in Eq. 9. This keeps the global path consistent with the neutral motion while allowing the remaining body features to become progressively more expressive.

Hyperparameters Default Values. The training-free approach uses 15 uniform keyframes and N_w=50 basis functions for the DMP-exaggerated motion (Q_K), and guidance weight ω=0.3 for the iterative refinement process. Text classifier-free guidance (CFG) scale is 2.5 for MDM, 5.0 for HY-Motion, and 2.5 for KIMODO.

D Remarks on Training-Time Exaggeration Approach

D.1 Supervised Fine-Tuning Formulation for HY-Motion

HY-Motion (Wen et al., 2025) is a flow-matching text-to-motion model. To formulate our training-time exaggeration approach (SFT) for this model, let $Q^{(\ell)} \in \mathbb{R}^{T \times D_{\text{H}}}$ denote a motion at exaggeration level ℓ after conversion to HY-Motion's normalized native representation, where $D_{\text{H}} = 201$. Given $Q^{(\ell)}$, we sample Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ and a continuous flow time $t \sim \mathcal{U}[0, 1]$, and construct the intermediate noisy state

$$z_t = (1 - t)\epsilon + tQ^{(\ell)}. \tag{25}$$

Under this linear flow path, the target (ground truth) velocity is

$$V^{(\ell)} = Q^{(\ell)} - \epsilon. \tag{26}$$

Let c_ℓ be the text condition describing the action and its exaggeration level. HY-Motion's diffusion transformer (DiT) (Peebles & Xie, 2023), $\mathcal{V}_\theta$, predicts the velocity

$$\widehat{V}^{(\ell)} = \mathcal{V}_\theta(z_t, t, c_\ell). \tag{27}$$

Then, we fine-tune the model using its native masked flow-matching objective:

$$\mathcal{L}_{\mathrm{SFT}}^{\mathrm{HY}} = \mathbb{E}_{Q^{(\ell)},t,\epsilon}\left[\frac{1}{D_{\mathrm{H}}\sum_{i=1}^{T} M_i}\sum_{i=1}^{T}\sum_{d=1}^{D_{\mathrm{H}}} M_i\left(V^{(\ell)} - \mathcal{V}_\theta(z_t,t,c_\ell)\right)_{i,d}^2\right], \tag{28}$$

where $M_i \in \{0,1\}$ is a temporal validity mask: $M_i = 1$ if frame i belongs to the original motion sequence and $M_i = 0$ if it was added as padding when batching variable-length motions. Consequently, padded frames do not contribute to the loss, and the denominator $D_{\mathrm{H}}\sum_{i=1}^{T} M_i$ averages the squared error over only the valid frame–feature entries.

In this way, exaggeration is introduced through text conditioning c_ℓ. We also apply classifier-free text dropout during fine-tuning so that classifier-free guidance remains available at inference time.

D.2 Supervised Fine-Tuning Formulation for KIMODO

KIMODO (Rempe et al., 2026) is a diffusion-based motion model with a two-stage denoiser and a clean-motion (x_0) prediction objective. Therefore, its supervised fine-tuning formulation follows the same forward-diffusion process as MDM in Eq. 1 (Sec. 3.2). Again, let $Q^{(\ell)} \in \mathbb{R}^{T\times D_{\mathrm{K}}}$ denote a motion at exaggeration level ℓ after conversion to KIMODO's normalized native representation, where $D_{\mathrm{K}} = 273$. Therefore, at diffusion step s, the intermediate noisy motion latent is

$$z_s = \sqrt{\bar{\alpha}_s}\,Q^{(\ell)} + \sqrt{1-\bar{\alpha}_s}\,\epsilon, \qquad \epsilon \sim \mathcal{N}(0,I). \tag{29}$$

Given the action-exaggeration text condition c_ℓ, KIMODO predicts the clean motion directly:

$$\widehat{Q}^{(\ell)} = \mathcal{G}_\theta(z_s, s, c_\ell). \tag{30}$$

The main difference from the MDM supervised fine-tuning formulation in Sec. 3.2 is the reconstruction objective. KIMODO partitions its 273-dimensional representation into six feature blocks $\mathcal{B}$:

$$\mathcal{B} = \begin{Bmatrix} b_{\mathrm{root}} & (3), & b_{\mathrm{heading}} & (2), & b_{\mathrm{pos}} & (66), \\ b_{\mathrm{rot}} & (132), & b_{\mathrm{vel}} & (66), & b_{\mathrm{contact}} & (4) \end{Bmatrix}. \tag{31}$$

These blocks respectively encode the smoothed root position b_{root}, the global root heading in a two-dimensional cosine–sine representation b_{heading}, the positions of 22 joints b_{pos}, the global 6D rotations of 22 joints b_{rot}, their velocities b_{vel}, and four foot-contact channels b_{contact}. Their dimensions sum to $3 + 2 + 66 + 132 + 66 + 4 = 273$.

KIMODO applies an element-wise Smooth-L_1 loss, also known as the Huber loss (Huber, 1992), to each feature block. For the normalized scalar residual $e_{i,d} = \widehat{Q}_{i,d}^{(\ell)} - Q_{i,d}^{(\ell)}$, the unit-threshold loss used in our implementation is

$$\mathrm{SmoothL1}(e_{i,d}) = \begin{cases} \frac{1}{2}e_{i,d}^2, & |e_{i,d}| < 1, \\ |e_{i,d}| - \frac{1}{2}, & |e_{i,d}| \geq 1. \end{cases} \tag{32}$$

It is quadratic for small reconstruction errors and linear for large errors, making it less sensitive to large residuals than a purely squared-error objective.

Let $|b|$ denote the dimensionality of feature block b. The masked blockwise reconstruction objective is

$$\mathcal{L}_{\mathrm{SFT}}^{\mathrm{KIM}} = \mathbb{E}_{Q^{(\ell)},s,\epsilon}\left[\sum_{b\in\mathcal{B}}\frac{1}{|b|\sum_{i=1}^{T} M_i}\sum_{i=1}^{T}\sum_{d\in b} M_i\,\mathrm{SmoothL1}\left(\widehat{Q}_{i,d}^{(\ell)} - Q_{i,d}^{(\ell)}\right)\right]. \tag{33}$$

Here, M_i is the temporal validity mask defined above: it equals 1 for valid motion frames and 0 for padded frames. Dividing each block loss by $|b|$ prevents higher-dimensional blocks, such as joint rotations, from dominating the objective solely because they contain more features. The block losses are equally weighted in our implementation.

For both HY-Motion and KIMODO, we use the LoRA and prior-preservation strategy introduced in the main paper (Sec. 3.2). The pre-trained backbone weights remain frozen under LoRA, and the

prior-preservation term applies the corresponding native objective (flow-matching velocity regression for HY-Motion and clean-motion reconstruction for KIMODO) to neutral motions generated by the frozen base model. This regularizes the learned exaggeration behavior toward the pre-trained motion prior.

D.3 Motion Representations and Backbones

The three backbones (MDM, KIMODO, and HY-Motion) operate on per-frame motion features, but differ in their feature dimensions, skeleton conventions, frame rates, and generative objectives. Therefore, we fine-tune and guide each model directly in its native representation. For the shared text-motion evaluation, motions generated by HY-Motion and KIMODO are decoded to 22-joint SMPL positions (Loper et al., 2015) and re-encoded in the HumanML3D 263-dimensional representation. Model-specific exaggeration control and physical metrics are computed from the corresponding native outputs. Table 3 summarizes the native representations used in our implementation.

The captured motions in ExMoCap are first retargeted to a common SMPL-22 body skeleton (Loper et al., 2015) and then encoded separately for each backbone. MDM uses the official HumanML3D representation (Guo et al., 2022), whereas HY-Motion and KIMODO use their native SMPL-H- and SMPL-X-based representations, respectively. In all cases, we normalize the resulting features using the statistics associated with the corresponding pre-trained checkpoint. This preserves the feature scale expected by each backbone during fine-tuning and inference.

MDM / HumanML3D-263. MDM (Tevet et al., 2023) is a transformer-encoder diffusion model with a frozen CLIP text encoder and a clean-motion (x_0) prediction objective. In our configuration, it uses a cosine noise schedule with 50 diffusion steps and a text-conditioning dropout probability of 0.1. Motion is represented at 20 FPS using the 263-dimensional HumanML3D (Guo et al., 2022) feature vector containing root angular velocity (1), root linear velocity in the ground plane (2), root height (1), root-relative positions of the 21 non-root joints (63=21×3), continuous 6D rotations of the 21 non-root joints (126=21×6) (Zhou et al., 2019), local velocities of all 22 joints (66=22×3), and four foot-contact features.

HY-Motion / SMPL-H-201. HY-Motion (Wen et al., 2025) is a flow-matching DiT that predicts a velocity field $v_\theta(x_t, c, t)$. Given Gaussian noise x_0 and clean motion x_1, our SFT implementation uses the linear optimal-transport path $x_t = (1-t)x_0+tx_1$ and the target velocity x_1-x_0. HY-Motion represents motion at 30 FPS using a 201-dimensional feature vector containing root translation (3), root orientation in 6D (6), local 6D rotations for the 21 non-root body joints (126=21×6), and auxiliary positions for the 22 body joints (66=22×3). We use the released Lite (0.46B) and Standard (1.0B) variants. Text conditioning combines a global 768-dimensional CLIP-L (Radford et al., 2021) embedding with 4096-dimensional per-token Qwen3-8B (Yang et al., 2025) features. At inference, we integrate the learned velocity field using 50 Euler steps and a text classifier-free-guidance scale of 5.0.

KIMODO / SMPL-X-273. KIMODO (Rempe et al., 2026) is a two-stage kinematic diffusion model: a root stage models a 5-dimensional global-root block, and a body stage models the remaining 268 dimensions. The model predicts clean motion (x_0) under a cosine DDPM schedule (Ho et al., 2020) with T=1000 training timesteps. Our inference uses 100-step DDIM sampling (Song et al., 2020). We use the body-only SMPL-X variant, which represents 22 joints at 30 FPS with a 273-dimensional feature vector containing a smoothed root position (3), global root heading represented by $[\cos, \sin]$ (2), joint-position features (66=22×3), global 6D joint rotations (132=22×6), joint velocities (66=22×3), and four foot-contact features. In the joint-position block, the horizontal coordinates are expressed relative to the smoothed root, while the vertical coordinate remains global. The checkpoint applies separate normalization statistics to the root block ([0:5]) and the body block ([5:273]). Text conditioning uses a 4096-dimensional LLM2Vec embedding (BehnamGhader et al., 2024) obtained from Meta-Llama-3-8B, and we use a text classifier-free-guidance scale of 2.5 at inference.

Text Conditioning. Each motion clip's caption combines an action phrase with an intensity phrase. The five intensity phrases are "in a natural way" (ℓ=0), "slightly exaggerated" (1), "moderately exaggerated" (2), "very exaggerated" (3), and "extremely exaggerated" (4); e.g. "a person kicks, very exaggerated". To diversify supervision we build up to four captions per clip (one canonical phrasing plus up to three synonym combinations drawn from small action- and intensity-synonym lists) and sample one per training epoch. Therefore, exaggeration enters purely through the text

Table 3: Native motion representations and generative parameterizations used by the three backbones including MDM, HY-Motion, and KIMODO. Motion feature dimensions are reported per frame.

	MDM (HumanML3D)	**HY-Motion** (SMPL-H)	**KIMODO** (SMPL-X)
Feature dimension	263	201	273
Body joints	22	22	22
Frame rate	20 FPS	30 FPS	30 FPS
Prediction target	clean motion (x_0)	velocity (v)	clean motion (x_0)
Text encoder	CLIP	CLIP-L + Qwen3-8B	LLM2Vec (Llama-3-8B)
Training / sampling	cosine DDPM, 50 steps	OT flow, 50 Euler steps	cosine DDPM (T=1000), 100-step DDIM
Per-frame feature layout (dimensions)			
Root features	angular vel. 1, planar vel. 2, height 1	translation 3, orientation 6	position 3, heading 2
Joint-position features	63	66	66
Joint rotations (6D)	126 (local)	126 (local)	132 (global)
Joint velocities	66	—	66
Foot contacts	4	—	4

prompt, so at inference the user communicates the exaggeration intensity by changing the phrase while keeping the action fixed.

D.4 Data Augmentation

Our graded-exaggeration dataset, ExMoCap, is limited, so we expand it using three label-preserving augmentations. To this end, we first apply each augmentation to the common SMPL-22 joint trajectories (Loper et al., 2015) before converting the resulting motion separately into the native representation of each motion model backbone. Specifically, we use the same backbone-specific preprocessing and feature-extraction pipeline used to construct its unaugmented training samples. This preserves the structural relationships among derived features, such as rotations, velocities, and foot-contact labels, which could become inconsistent if the native feature vectors were modified directly. In the following, we explain the three data augmentation techniques used to expand our ExMoCap-dataset.

Left–Right Mirroring. Following the standard augmentation used in HumanML3D (Guo et al., 2022), we reflect each motion across the sagittal plane and exchange the corresponding left and right joints. This produces an anatomically consistent mirrored motion while preserving its action, exaggeration level, and text condition. We generate one mirrored variant of each motion window in the dataset.

Position Jitter. We generate an additional variant by adding small zero-mean Gaussian perturbations to the joint positions, using a standard deviation of 1 cm. The perturbations are smoothed temporally to suppress high-frequency noise and preserve the overall action and exaggeration level.

Balanced Windowing. To prevent longer motion clips from contributing disproportionately more training samples, we draw approximately 16 distinct temporal windows from each motion clip. Window durations are sampled uniformly between 2.0 and 9.8 seconds, corresponding to 40–196 frames for MDM at 20 FPS and 60–294 frames for HY-Motion and KIMODO at 30 FPS. Motion clips shorter than the minimum window duration are kept as a single full-length window. Each selected window contributes its original, mirrored, and position-jittered variants.

D.5 Implementation Details

We initialize each model from its public pre-trained checkpoint and retain its native architecture, motion representation, and generative objective. All losses are computed in the corresponding normalized feature space, and padded frames are excluded through temporal masks. Unless otherwise specified, we use the LoRA-based fine-tuning strategy described below and full fine-tuning is used only for the corresponding ablation in Tables 6-9.

Low-Rank Adaptation. For a pre-trained linear layer with weight matrix W, LoRA (Hu et al., 2021) introduces a trainable residual $\Delta W = (\alpha/r)BA$, where r is the adapter rank. We initialize A using Kaiming-uniform initialization (He et al., 2015), which scales its random weights according to the input dimension, and initialize B to zero. Consequently, $\Delta W = 0$ at initialization, so the

Table 4: **Default supervised fine-tuning hyperparameters and configurations.**

Setting	MDM	HY-Motion	KIMODO
Objective	masked x_0 MSE	masked velocity MSE	masked x_0 smooth-L_1
Training steps	16k	16k	16k
Batch size / GPU	64	16	8
Learning rate	1×10^{-4}	1×10^{-4}	1×10^{-4}
Maximum frames	196	294	300
LoRA rank r	$\{64, 128\}$	$\{16, 64\}$	16
LoRA scale α	r	16	16
CFG dropout	0.1	0.1	0.1
Gradient clipping	1.0	1.0	1.0
EMA decay	0.9999	0.999	0.999
λ_{prior}	$\{0.1, 0.25\}$	$\{0.1, 0.25\}$	0.25
Prior source	HumanML3D	frozen-base samples	frozen-base samples
Precision	FP32	FP32	FP32

adapted model initially behaves identically to the pre-trained model. The pre-trained weights remain frozen, and only the adapter parameters are optimized. The adapted layers follow the structure of each backbone: MDM adapts the query, key, and value projections in the denoising transformer's attention layers; HY-Motion adapts the attention and MLP projections of its DiT; and KIMODO adapts the transformer feed-forward layers together with its motion input, motion output, and text projections. MDM sets $\alpha = r$. HY-Motion and KIMODO use $\alpha = 16$, which gives unit scaling at their default rank $r = 16$.

Prior Preservation. To reduce forgetting of the pre-trained motion distribution, we augment the supervised exaggeration loss with a prior-preservation regularization term,

$$\mathcal{L}_{\text{SFT}} = \mathcal{L}_{\text{exag}} + \lambda_{\text{prior}}\mathcal{L}_{\text{prior}}. \tag{34}$$

The prior term applies the same native training objective as the exaggeration term: clean-motion denoising for MDM (Eq. 2) and KIMODO (Eq. 33), and velocity regression for HY-Motion (Eq. 28). For MDM, the reported experiments use real HumanML3D motions Guo et al. (2022) as prior examples. For HY-Motion and KIMODO, we generate a fixed set of neutral motions using the corresponding frozen base model and cache them before fine-tuning.

Optimization. We optimize with AdamW (Loshchilov & Hutter, 2017) and apply classifier-free-conditioning dropout with probability 0.1 using each backbone's native null condition. Gradients are clipped to a maximum norm of 1.0. We maintain an exponential moving average of the trainable parameters for validation and checkpoint sampling, using decay 0.9999 for MDM and 0.999 for HY-Motion and KIMODO. Table 4 summarizes the default configurations used by our experiments.

E Additional Experiments and Results

We show expanded results, ablation studies, and discussions from the main paper featuring more controllable motion exaggeration examples across more diverse actions using our different approaches including TF, SFT, and SFT+TF in Figs. 5- 11, and Tables. 6- 10.

Qualitative Results on More Diverse Actions. We provide additional qualitative results to complement the representative comparisons in the main paper. First, Figs. 6–8 separately visualize motions generated using our SFT+TF, SFT, and TF approaches across a broader set of actions. These comparisons illustrate how each approach increases the expressiveness of the generated motion while retaining the structure of the underlying action. Second, Fig. 9 extends the qualitative evaluation to additional examples generated with the MDM and HY-Motion backbones, demonstrating that the proposed exaggeration framework is applicable across different motion representations and generative architectures. Finally, Figs. 10 and 11 illustrate the controllability of the proposed approaches. For SFT, exaggeration intensity is controlled through the intensity description included in the text

Table 5: **User-study results for controllable exaggeration and action preservation.** We report how often participants identified the highest requested intensity as the most exaggerated and the neutral reference motion as the most neutral. Action preservation is the percentage of ratings marked *very well preserved* or *mostly preserved*. Percentages are computed over valid responses, with counts shown in parentheses; "–" indicates that no example was evaluated.

Backbone	Exag. actions	Highest intensity	Neutral reference	Preserv. actions	Action preserved
KIMODO	3	86.7% (52/60)	80.0% (48/60)	5	82.0% (82/100)
HY-Motion	1	78.9% (15/19)	10.0% (2/20)	1	50.0% (10/20)
MDM	1	70.0% (14/20)	25.0% (5/20)	0	–
Overall	5	81.8% (81/99)	55.0% (55/100)	6	76.7% (92/120)

condition, whereas for TF, it is controlled through the DMP exaggeration scale K. In both cases, increasing the corresponding control produces progressively more pronounced poses and motion amplitudes while preserving the identity of the original action.

Ablation studies on training-time (SFT) exaggeration approach. The purpose of these ablations is to identify, for each motion-generation backbone, an SFT configuration that provides a balanced trade-off between exaggeration control and action preservation metrics. We evaluate exaggeration control using $\mathrm{ELRA@1}$, ρ_{clf}, ρ_{phys}, and Δ_{phys} (explained in Appendix B), while action preservation and fidelity to the pre-trained motion distribution are assessed using R-precision, FID, Diversity, and MMDist following evaluation protocol used in (Guo et al., 2022). Additionally, Foot-Skate is reported to monitor motion quality. Across MDM, KIMODO, HY-Motion (0.4B), and HY-Motion (1B), we vary the fine-tuning strategy where applicable, the LoRA rank r, and the prior-preservation weight λ. For MDM, we additionally examine the source of the prior-preservation data. These factors control the effective adaptation strength. Specifically, greater adaptation capacity or weaker prior preservation can improve exaggeration control but may cause the generated motion to deviate from the original action and the pre-trained motion distribution, whereas stronger prior preservation can retain the original motion prior at the cost of suppressing exaggeration. Therefore, we do not select configurations by maximizing any single metric and instead we choose a balanced operating point that substantially improves exaggeration controllability while limiting degradation in action preservation and motion quality. Tables 6-9 report the complete results used for this selection.

User Study. To perceptually assess the exaggeration capability of our method and its ability to preserve the underlying action, we conducted a user study with 20 participants. The controllable-exaggeration evaluation comprised 5 videos, each showing a neutral reference motion (level 0) alongside three or four variants generated by our method at progressively increasing exaggeration levels. The walking, dancing, and kicking, leaping-forward, and jumping example used for this evaluation. The display order was permuted across examples, and the characters were identified only by letters so that the underlying exaggeration levels were not disclosed. For each video, participants selected both the most exaggerated motion and the most neutral motion, resulting in 10 questions per participant. We report the *highest-intensity selection rate*, defined as the percentage of valid responses in which the largest requested exaggeration level was selected as the most exaggerated, and the *neutral-reference selection rate*, defined as the percentage in which neutral reference motion was selected as the most neutral. For the *action-preservation* evaluation, participants viewed a neutral source motion and its exaggerated counterpart and rated whether the counterpart preserved the type of action performed by the source character. The response options were *very well preserved*, *mostly preserved*, *slightly preserved*, and *not preserved*. We count the first two categories as positive responses. This evaluation comprised 6 questions. In total, each participant was presented with 16 questions. Fig. 12 illustrates representative questions from the controllable-exaggeration and action-preservation evaluations presented to participants. As reported in Table 5, participants selected the highest requested exaggeration level as the most exaggerated in 81.8% of valid responses and selected the neutral reference motion as the most neutral in 55% of responses. The exaggerated motions received a positive action-preservation rating in 76.7% of cases. These results indicate that the highest requested exaggeration level was generally perceived as the most exaggerated while the underlying action was preserved in most evaluations.

F THE USE OF LARGE LANGUAGE MODELS

We used large language models (LLMs) for the following purposes: (i) editing, refining, and polishing the manuscript, including improving its grammar, wording, concision, organization, and overall readability; (ii) supporting retrieval and discovery during the literature review, particularly when searching for existing motion foundation models and prior work related to animation principles; and (iii) assisting with the presentation and verification of the work, including checking the consistency between manuscript descriptions and our implementation, and developing a script to aggregate the user-study responses and format the resulting statistics.

LLMs were not used to generate the motion datasets or experimental results, run the reported experiments, or make final methodological or scientific decisions. All LLM-generated suggestions, including text, literature leads, and code, were carefully reviewed, verified, and, revised by the authors. Relevant literature was checked against the original sources, and all reported analyses and results were validated by the authors. The authors take full responsibility for the final manuscript and the accuracy of all its content, claims, results, and artifacts.

Table 6: **Ablation study of training-time (SFT) exaggeration approach on MDM.** We evaluate the effect of (a) fine-tuning strategy, (b) LoRA rank, (c) prior-preservation weight, and (d) prior-preservation data source across metrics measuring exaggeration control, prompt adherence, motion-manifold fidelity, and motion quality. The goal is to identify effective hyperparameters and configurations for supervised fine-tuning of the MDM model. Section (a) compares two fine-tuning strategies: (i) *LoRA (ours)*, which trains LoRA adapters with the proposed prior-preservation objective, and (ii) *Full FT*, which updates all model parameters using the same objective. Full fine-tuning produces stronger exaggeration, but substantially degrades prompt adherence and motion-manifold fidelity. In contrast, LoRA with prior preservation provides a better trade-off between exaggeration control and preservation of the base model's action semantics and motion manifold. In Sections (b) and (c), we ablate this LoRA+prior-preservation configuration by training identical models with different LoRA ranks r and prior-preservation weights λ, respectively. Increasing the LoRA rank consistently improves exaggeration control but progressively reduces prompt adherence and manifold fidelity. Increasing the prior-preservation weight exhibits the opposite trend, improving prompt adherence and manifold fidelity while reducing the strength of exaggeration. Therefore, we select $r \in \{64, 128\}$ and $\lambda \in \{0.1, 0.25\}$, providing a practical trade-off between exaggeration and preservation. Finally, Section (d) compares different sources for the prior-preservation data: real HumanML3D motions, motions generated by the pre-trained MDM, and a mixture of the two. Using real HumanML3D motions generally provides stronger preservation of prompt adherence and motion-manifold fidelity, while generated or mixed priors provide comparable but slightly weaker results.

	Exaggeration				**Action Preservation**						**Quality**
Setting	ELRA@1 ↑	ρ_{clf} ↑	ρ_{phys} ↑	Δ_{phys} ↑	R@1↑	R@2↑	R@3↑	FID↓	Div→	MMDist↓	Skate↓
Reference rows											
ExMoCap	0.983	0.995	0.564	1.494	–	–	–	–	–	–	0.050
HumanML3D	–	–	–	–	0.4530	0.6597	0.7675	0.0010	9.2461	3.2417	0.044
Base MDM (no FT)	0.199	0.014	0.099	0.273	0.4019	0.6062	0.7211	0.4444	9.5169	3.5280	0.076
(a) Fine-tuning strategy											
LoRA (ours)	0.227	0.165	0.393	1.139	**0.3929**	**0.5882**	**0.6990**	**0.5652**	9.4828	**3.6276**	0.086
Full FT	**0.342**	**0.530**	**0.744**	**2.160**	0.2973	0.4520	0.5522	3.7578	8.9353	4.5992	**0.082**
(b) LoRA rank											
r=1	0.203	0.036	0.120	0.318	0.4004	**0.6025**	**0.7173**	0.4747	9.4540	**3.5344**	0.087
r=4	0.202	0.009	0.144	0.365	0.4007	0.6000	0.7143	0.4616	9.3847	3.5560	0.094
r=8	0.208	0.043	0.265	0.732	**0.4038**	0.6002	0.7132	**0.3833**	9.3952	3.5489	0.095
r=16	0.219	0.097	0.369	1.075	0.4035	0.5975	0.7075	0.4441	9.4274	3.5787	0.090
r=32	0.232	0.182	0.421	1.212	0.3905	0.5828	0.6949	0.6024	9.3899	3.6576	0.084
r=64	0.256	0.276	0.442	1.257	0.3755	0.5602	0.6745	0.8684	9.3040	3.7732	0.078
r=128	0.301	0.370	0.511	1.522	0.3547	0.5405	0.6494	1.2379	9.2199	3.9219	0.072
r=256	0.326	0.423	0.586	1.738	0.3346	0.5168	0.6242	1.8154	9.1292	4.1080	0.069
r=512	0.339	0.433	0.646	1.879	0.3161	0.4855	0.5923	2.6873	8.9966	4.3296	0.066
r=768	0.340	0.447	0.676	1.995	0.3004	0.4653	0.5709	3.3625	8.9418	4.4975	0.065
r=1024	**0.357**	**0.477**	**0.710**	**2.058**	0.2916	0.4530	0.5590	3.4683	8.8735	4.5707	**0.061**
(c) Prior-preservation weight											
λ=0.01	**0.367**	**0.500**	**0.734**	**2.150**	0.2630	0.4108	0.5121	4.6972	8.7262	4.9157	**0.059**
λ=0.03	0.348	0.469	0.709	2.090	0.2816	0.4376	0.5411	4.0762	8.8212	4.7135	0.060
λ=0.05	0.357	0.477	0.710	2.058	0.2916	0.4530	0.5590	3.4683	8.8735	4.5707	0.061
λ=0.1	0.362	0.482	0.700	2.033	0.3042	0.4697	0.5752	3.2138	8.9436	4.4339	0.063
λ=0.25	0.346	0.439	0.687	2.000	0.3217	0.4958	0.6020	2.5860	8.9760	4.2150	0.065
λ=0.5	0.333	0.401	0.656	1.915	0.3445	0.5225	0.6333	1.9261	9.0553	4.0145	0.066
λ=0.75	0.324	0.362	0.626	1.842	0.3553	0.5419	0.6500	1.5578	9.1295	3.9030	0.067
λ=1	0.311	0.336	0.611	1.753	0.3652	0.5505	0.6616	1.3489	9.1905	3.8341	0.070
λ=5	0.240	0.098	0.471	1.401	**0.3926**	**0.5875**	**0.7027**	**0.7383**	9.7111	**3.6403**	0.091
(d) Prior-preservation source											
HumanML3D	**0.217**	**0.142**	**0.410**	**1.211**	**0.3831**	**0.5736**	**0.6873**	1.3971	9.0649	**3.6922**	**0.086**
Generated	0.213	0.112	0.312	0.963	0.3690	0.5546	0.6671	1.6654	9.1062	3.8199	0.089
Mixed	0.217	0.112	0.334	1.031	0.3804	0.5683	0.6824	**1.3573**	9.1361	3.7221	0.090

Table 7: **Ablation study of Training-time (SFT) exaggeration approach on HY-Motion (0.4B).** We evaluate the effect of (a) fine-tuning strategy, (b) LoRA rank, and (c) prior-preservation weight across metrics measuring exaggeration control, prompt adherence, motion-manifold fidelity, and motion quality. The goal is to identify effective hyperparameters and configurations for supervised fine-tuning of the HY-Motion (0.4B) model. Section (a) compares two fine-tuning strategies: (i) *LoRA (ours)*, which trains LoRA adapters with the proposed prior-preservation objective, and (ii) *Full FT*, which updates all model parameters without regularization. Full fine-tuning produces substantially stronger exaggeration but severely degrades prompt adherence and motion-manifold fidelity. In contrast, LoRA with prior preservation significantly improves exaggeration control while retaining the pretrained model's action semantics and motion manifold. In Section (b), we ablate the LoRA rank r while keeping the prior-preservation weight fixed. Increasing r improves exaggeration control from $r = 1$ up to approximately $r = 16$, after which the gains largely saturate. Meanwhile, prompt-adherence and manifold metrics remain relatively stable across ranks, with no consistent degradation as observed for MDM. We therefore select a relatively small rank, $r = 16$, for HY-Motion (0.4B), which provides strong exaggeration control. In Section (c), we ablate the prior-preservation weight λ. Removing prior preservation ($\lambda = 0$) yields the strongest classifier-based exaggeration, but substantially worsens prompt adherence and manifold fidelity. Increasing λ generally improves prompt adherence, with the strongest R-precision obtained around $\lambda \in \{0.25, 0.5\}$, while larger values eventually weaken exaggeration control. We therefore use $\lambda = 0.25$ for the selected configuration, providing a strong balance between exaggeration and preservation of the pretrained motion prior.

	Exaggeration				**Action Preservation**						**Quality**
Setting	ELRA@1 ↑	ρ_{clf} ↑	ρ_{phys} ↑	Δ_{phys} ↑	R@1↑	R@2↑	R@3↑	FID↓	Div→	MMDist↓	Skate↓
Reference rows											
ExMoCap	0.995	0.999	0.542	1.534	–	–	–	–	–	–	0.071
HumanML3D	–	–	–	–	0.4516	0.6596	0.7685	0.0010	9.0560	3.2341	–
Base HY-Motion (no FT)	0.335	0.453	0.335	0.970	0.3004	0.4617	0.5827	3.5259	8.1049	4.5667	0.046
(a) Fine-tuning strategy											
LoRA (ours)	0.763	0.943	0.531	1.491	**0.2883**	**0.4516**	**0.5706**	**4.7590**	8.1070	**4.6841**	0.061
Full FT	**0.962**	**0.989**	**0.539**	**1.549**	0.1724	0.2823	0.3679	12.6007	7.5104	6.1947	**0.060**
(b) LoRA rank											
r=1	0.653	0.889	0.552	1.568	0.3054	0.4657	0.5776	**4.1510**	7.8760	4.5865	**0.054**
r=4	0.700	0.915	0.559	1.562	**0.3135**	**0.4778**	0.5847	4.2825	7.9836	**4.5031**	0.061
r=8	0.738	0.931	0.527	1.502	0.3014	0.4748	**0.5877**	4.5399	7.8131	4.6309	0.059
r=16	**0.768**	**0.942**	0.501	1.444	0.2933	0.4758	0.5756	4.6824	8.1263	4.6606	0.058
r=32	0.748	0.938	**0.561**	**1.586**	0.2873	0.4698	0.5806	4.5664	7.9419	4.6014	0.057
r=64	0.765	0.933	0.523	1.485	0.2923	0.4667	0.5756	4.6874	8.1852	4.6542	0.061
r=128	0.752	0.934	0.523	1.466	0.2944	0.4516	0.5504	4.4536	8.1859	4.6502	0.058
r=256	0.752	0.931	0.530	1.529	0.2903	0.4587	0.5706	4.6146	7.9727	4.6222	0.059
r=512	0.703	0.931	0.539	1.525	0.2954	0.4667	0.5817	4.3785	8.4534	4.6506	0.059
(c) Prior-preservation weight											
λ=0	**0.839**	**0.955**	0.527	1.494	0.2782	0.4284	0.5252	6.3173	8.1907	5.1383	0.059
λ=0.05	0.737	0.931	0.530	1.504	0.2933	0.4718	0.5867	4.4554	7.9255	4.5993	0.059
λ=0.1	0.736	0.933	0.527	1.475	0.2994	0.4526	0.5726	4.9073	8.1206	4.6936	0.058
λ=0.25	0.738	0.938	0.533	1.507	0.3135	0.4788	0.5817	**4.3490**	7.9348	**4.5337**	0.059
λ=0.5	0.724	0.923	0.526	1.485	**0.3145**	0.4788	0.5857	4.4406	8.0837	4.6461	**0.054**
λ=1	0.691	0.925	**0.546**	**1.533**	0.3125	0.4788	0.5857	4.5824	8.2268	4.6596	0.057
λ=5	0.527	0.833	0.502	1.407	0.3054	**0.4909**	**0.6109**	4.5394	8.3926	4.6541	0.058

Table 8: **Ablation study of training-time (SFT) exaggeration approach on HY-Motion (1B).** We evaluate the effect of (a) fine-tuning strategy, (b) LoRA rank, and (c) prior-preservation weight across metrics measuring exaggeration control, prompt adherence, motion-manifold fidelity, and motion quality. The goal is to identify effective hyperparameters and configurations for supervised fine-tuning of the HY-Motion (1B) model. Section (a) compares two fine-tuning strategies: (i) *LoRA (ours)*, which trains LoRA adapters with the proposed prior-preservation objective, and (ii) *Full FT*, which updates all model parameters without regularization. Full fine-tuning produces stronger exaggeration, but significantly degrades prompt adherence and motion-manifold fidelity. In contrast, LoRA with prior preservation maintains much stronger prompt adherence and manifold fidelity while still providing substantial exaggeration control. In Section (b), we ablate the LoRA rank r with a fixed prior-preservation weight. Increasing r generally improves classifier-based exaggeration control up to $r = 128$. Prompt adherence and manifold fidelity also remain relatively stable across the tested ranks, with $r = 256$ providing the strongest R-precision and MMDist among the rank ablations. We therefore select $r = 64$ for the HY-Motion (1B) configuration, providing strong exaggeration control while retaining competitive prompt adherence and manifold fidelity without the additional capacity of larger adapters. In Section (c), we ablate the prior-preservation weight λ. Removing prior preservation ($\lambda = 0$) gives the strongest classifier-based exaggeration, but significantly reduces prompt adherence and increases FID and MMDist. Introducing a small prior-preservation weight improves prompt adherence and manifold fidelity, with $\lambda = 0.25$ providing particularly strong R-precision together with a high physical-intensity gap, while larger weights progressively weaken classifier-based exaggeration. We therefore use $\lambda = 0.25$ for the selected configuration, providing a practical balance between exaggeration control and preservation of the pretrained motion prior.

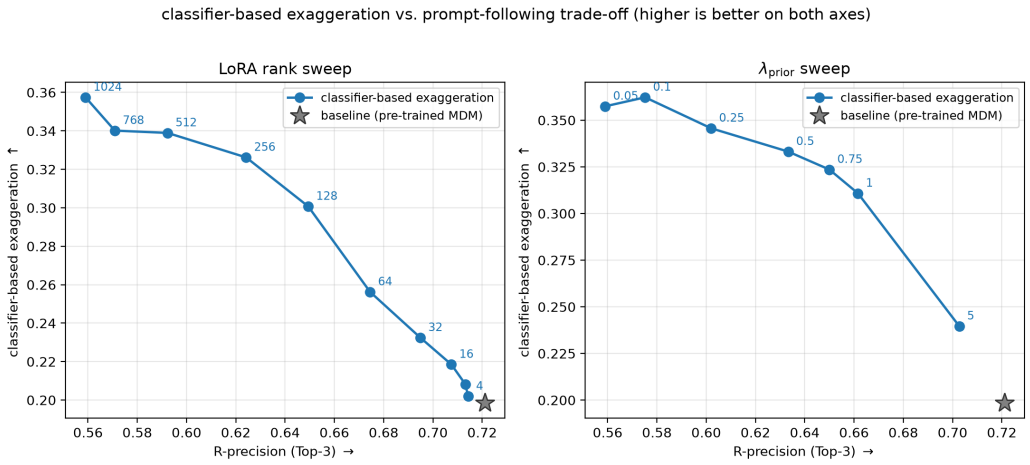
	Exaggeration				**Action Preservation**						**Quality**
Setting	ELRA@1 ↑	ρ_{clf} ↑	ρ_{phys} ↑	Δ_{phys} ↑	R@1↑	R@2↑	R@3↑	FID↓	Div→	MMDist↓	Skate↓
Reference rows											
ExMoCap	0.995	0.999	0.542	1.534	–	–	–	–	–	–	0.071
HumanML3D	–	–	–	–	0.4527	0.6619	0.7719	0.0010	9.4287	3.2242	–
Base HY-Motion 1B (no FT)	0.291	0.346	0.300	0.720	0.3266	0.5262	0.6351	2.9705	8.2249	4.3312	0.044
(a) Fine-tuning strategy											
LoRA (ours)	0.841	0.953	**0.542**	1.554	**0.3276**	**0.5242**	**0.6310**	**3.5592**	**8.4329**	**4.3457**	0.063
Full FT	**0.944**	**0.986**	0.538	**1.562**	0.1734	0.3145	0.4032	5.8215	7.9433	5.5796	**0.038**
(b) LoRA rank											
r=1	0.730	0.920	**0.609**	**1.740**	0.3135	0.4909	0.5998	3.7862	7.8361	4.4610	0.062
r=4	0.792	0.946	0.559	1.570	0.3347	0.5192	0.6230	3.3686	8.5369	4.4019	0.062
r=8	0.818	0.955	0.547	1.543	0.3357	0.4889	0.6038	3.4149	8.2937	4.3819	0.060
r=16	0.813	0.941	0.535	1.523	0.3226	0.4879	0.6069	**3.3473**	8.0569	4.4192	0.063
r=32	0.835	0.948	0.542	1.548	0.3145	0.4909	0.6129	3.6444	8.1022	4.4219	0.062
r=64	0.850	0.957	0.521	1.509	0.3196	0.5121	**0.6300**	3.5303	8.1566	4.3980	0.062
r=128	**0.855**	**0.963**	0.521	1.484	0.3276	0.4970	0.5938	3.5654	8.2790	4.4507	**0.058**
r=256	0.844	0.954	0.540	1.544	**0.3488**	**0.5212**	0.6250	3.4028	8.4173	**4.3071**	0.064
(c) Prior-preservation weight											
λ=0	**0.889**	**0.972**	0.536	1.526	0.2792	0.4315	0.5282	4.9065	8.2334	4.8583	0.065
λ=0.05	0.822	0.954	0.543	1.542	0.3145	0.4990	0.6099	3.4945	8.4698	4.4523	0.063
λ=0.1	0.833	0.949	0.540	1.536	0.3165	0.5020	0.6139	**3.3388**	8.0177	4.4065	0.062
λ=0.25	0.841	0.953	0.542	**1.554**	0.3276	**0.5242**	**0.6310**	3.5592	8.4329	4.3457	0.063
λ=0.5	0.806	0.945	0.531	1.522	0.3256	0.5212	0.6250	3.5808	8.4624	4.3903	**0.059**
λ=0.75	0.784	0.936	**0.548**	1.534	0.3135	0.5010	0.6290	3.5834	8.6047	4.3661	0.063
λ=1	0.803	0.942	0.545	1.532	0.3256	0.5091	0.6300	3.6054	8.4391	4.3395	0.063
λ=5	0.693	0.891	0.546	1.550	**0.3387**	0.5161	0.6260	3.4538	8.6014	**4.3095**	0.065

Table 9: **Ablation study of training-tree (SFT) exaggeration approach on KIMODO.** We evaluate the effect of (a) LoRA rank and (b) prior-preservation weight across metrics measuring exaggeration control, prompt adherence, motion-manifold fidelity, and motion quality. The goal is to identify effective hyperparameters and configurations for supervised fine-tuning of the KIMODO model. In Section (a), we ablate the LoRA rank r with a fixed prior-preservation weight. Unlike MDM and HY-Motion, in KIMODO, the effect of LoRA rank on exaggeration control is not monotonic: $\mathrm{ELRA@1}$ and ρ_{clf} improve from $r = 1$ to moderate ranks but fluctuate thereafter, while the strongest physical-intensity correlation and neutral-to-extreme physical gap are obtained at the largest tested rank, $r = 512$. Prompt adherence and manifold fidelity also vary non-monotonically across ranks, with different ranks providing the strongest individual metrics; for example, $r = 256$ gives the highest R@1, while $r = 32$ gives the lowest FID. These results suggest that increasing LoRA capacity does not uniformly improve either exaggeration control or prompt adherence for KIMODO. In Section (b), we ablate the prior-preservation weight λ. Removing prior preservation ($\lambda = 0$) gives the strongest classifier-based exaggeration metrics, but substantially degrades prompt adherence and manifold fidelity. Introducing a small prior-preservation weight improves the text-motion alignment metrics, with $\lambda \in \{0.1, 0.25\}$ providing strong R-precision and MMDist, while larger values generally reduce classifier-based exaggeration. The physical-intensity metrics remain relatively stable across moderate values of λ, with $\lambda = 0.5$ achieving the largest Δ_{phys}. Based on these results, we select a moderate LoRA rank and a small prior-preservation weight to balance exaggeration control with preservation of the pretrained motion prior.

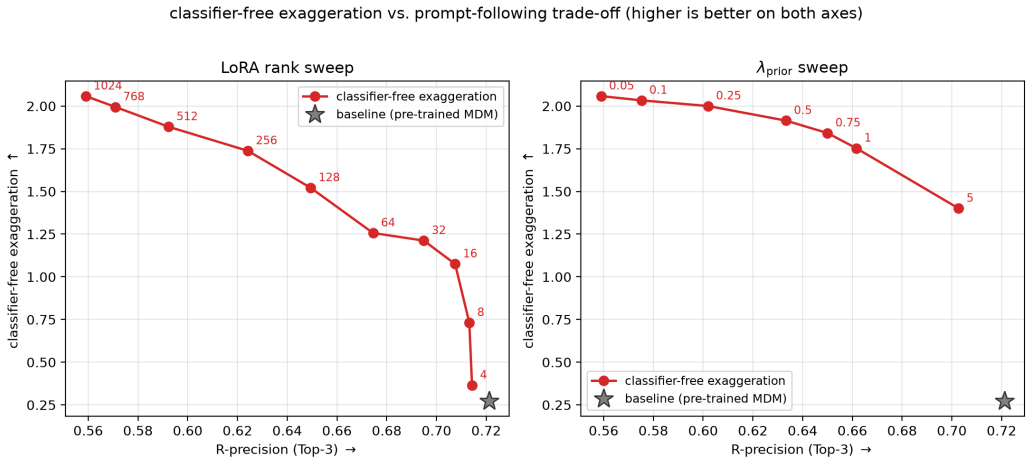
	Exaggeration				**Action Preservation**						**Quality**
Setting	ELRA@1 ↑	ρ_{clf} ↑	ρ_{phys} ↑	Δ_{phys} ↑	R@1↑	R@2↑	R@3↑	FID↓	Div→	MMDist↓	Skate↓
Reference rows											
ExMoCap	0.990	0.998	0.540	1.527	–	–	–	–	–	–	0.071
HumanML3D	–	–	–	–	0.4537	0.6611	0.7748	0.0010	9.4376	3.2301	–
Base KIMODO (no FT)	0.203	-0.024	0.174	0.448	0.3291	0.5373	0.6442	1.7365	8.9833	4.0029	0.026
(a) LoRA rank											
r=1	0.450	0.696	0.598	1.660	0.3377	0.5192	0.6290	1.3308	9.3574	**4.0667**	**0.032**
r=4	0.514	0.782	0.600	1.651	0.3387	**0.5252**	0.6240	1.2221	9.2667	4.0843	0.034
r=8	0.477	0.743	0.614	1.701	0.3306	0.5010	0.6038	1.3012	9.3138	4.1914	0.035
r=16	**0.527**	0.786	0.592	1.666	0.3306	0.5050	**0.6331**	1.2317	8.8285	4.1005	0.036
r=32	0.499	0.798	0.595	1.652	0.3306	0.5222	0.6159	**1.0634**	8.7674	4.0669	0.040
r=64	0.515	**0.827**	0.594	1.649	0.3427	0.4980	0.6280	1.1625	9.2137	4.1029	0.035
r=128	0.500	0.751	0.597	1.648	0.3337	0.5171	0.6139	1.2074	8.8401	4.0955	0.041
r=256	0.491	0.791	0.602	1.675	**0.3438**	0.5071	0.6139	1.2317	8.7661	4.1262	0.038
r=512	0.500	0.805	**0.615**	**1.710**	0.3347	0.4950	0.6149	1.2379	9.0761	4.1365	0.035
(b) Prior-preservation weight											
λ=0	**0.525**	**0.819**	**0.604**	1.655	0.3196	0.4677	0.5786	2.0470	9.0004	4.4328	0.042
λ=0.05	0.489	0.771	0.595	1.655	0.3256	0.5101	0.6179	1.1698	9.2136	4.1346	0.040
λ=0.1	0.492	0.744	0.600	1.653	0.3397	**0.5242**	0.6270	1.2117	8.9204	4.0824	**0.037**
λ=0.25	0.476	0.768	0.600	1.672	**0.3438**	0.5121	0.6280	1.1693	9.0286	**4.0539**	0.039
λ=0.5	0.498	0.744	0.599	**1.691**	0.3276	0.5040	0.6139	**1.1671**	8.9467	4.0736	0.040
λ=1	0.487	0.739	**0.604**	1.673	0.3226	0.5060	**0.6341**	1.3789	8.9422	4.0915	0.039
λ=5	0.412	0.687	0.569	1.615	0.3185	0.5091	0.6240	1.3913	9.0369	4.0880	0.042

Table 10: **Contribution of the proposed motion-exaggeration approaches.** We compare the original HY-Motion-0.4B, HY-Motion-1B, MDM, and KIMODO baselines with DMP, training-free guidance (TF), supervised fine-tuning (SFT), and their composition (SFT+TF). Each approach is evaluated using its corresponding exaggeration control: text-conditioned intensity levels for the Baseline and SFT, increasing DMP scale factors for DMP and TF, and both controls jointly for SFT+TF. The metrics evaluate exaggeration control, prompt adherence and motion-manifold fidelity, and motion quality, revealing the trade-off between stronger exaggeration and preservation of the underlying action. "ExMoCap" provides a reference for the exaggeration-control metrics, while "HumanML3D" provides a reference for the standard text-to-motion metrics; neither reference row is included when identifying the best method. ↑ and ↓ indicate that higher and lower values are better, respectively. The best result among the generated-motion methods for each backbone and metric is shown in bold.

		Exaggeration control					**Prompt adherence & manifold**						**Quality**
Model	**Method**	ELRA@1 ↑	ρ_{clf} ↑	ρ_{phys} ↑	Δ_{phys} ↑	Nov.↑	R@1↑	R@2↑	R@3↑	FID↓	Div↑	MMDist↓	Skate↓
HY-Motion-0.4B	ExMoCap	0.995	0.999	0.542	1.534	–	–	–	–	–	–	–	–
	HumanML3D	–	–	–	–	–	0.4481	0.6554	0.7640	0.0011	9.3215	3.2445	–
	Baseline	0.331	0.267	0.315	0.961	0.069	**0.3945**	**0.6016**	**0.7305**	**2.5235**	8.3203	**3.5885**	0.051
	+ DMP	0.231	0.055	0.471	1.257	0.185	0.1875	0.2969	0.3828	34.5822	4.6140	6.2699	0.080
	+ TF	0.223	0.039	0.366	0.947	0.107	0.2383	0.3867	0.5273	22.0843	5.9314	5.1676	0.072
	+ SFT	**0.923**	**0.982**	0.554	1.573	0.214	0.3320	0.5625	0.6836	3.0531	8.2520	3.9408	**0.042**
	+ SFT+TF	0.892	0.975	**0.708**	**1.997**	**0.309**	0.2227	0.3711	0.5039	21.1181	5.7263	5.2920	0.067
HY-Motion-1B	ExMoCap	0.995	0.999	0.542	1.534	–	–	–	–	–	–	–	–
	HumanML3D	–	–	–	–	–	0.4582	0.6631	0.7681	0.0010	9.4410	3.2359	–
	Baseline	0.354	0.354	0.391	0.902	0.135	**0.4414**	**0.6484**	**0.7422**	**2.0825**	8.9325	**3.4851**	**0.029**
	+ DMP	0.200	-0.042	0.516	1.318	0.167	0.1523	0.2969	0.4023	30.8515	4.7568	6.0145	0.061
	+ TF	0.200	0.007	0.424	1.017	0.082	0.2266	0.4336	0.5898	17.0509	6.4124	4.8573	0.057
	+ SFT	**0.938**	**0.985**	0.556	1.622	0.222	0.4102	0.6367	0.7383	2.2297	8.4439	3.6413	0.040
	+ SFT+TF	0.900	0.973	**0.697**	**2.003**	**0.301**	0.2812	0.4297	0.5430	19.7208	5.8073	5.1643	0.063
MDM	ExMoCap	0.983	0.995	0.564	1.494	–	–	–	–	–	–	–	0.050
	HumanML3D	–	–	–	–	–	0.4532	0.6558	0.7690	0.0011	9.2666	3.2277	0.044
	Baseline	0.200	0.000	0.026	0.175	0.117	**0.4102**	**0.5898**	0.7031	**0.8388**	9.5722	**3.6133**	**0.070**
	+ DMP	0.200	0.000	0.358	0.953	0.130	0.2734	0.4219	0.5625	20.9116	6.8088	5.2530	0.134
	+ TF	0.200	0.000	0.252	0.552	0.023	0.3594	0.5859	**0.7070**	5.7547	8.3935	3.9870	0.091
	+ SFT	**0.408**	0.635	0.399	1.173	0.168	0.2969	0.4844	0.5547	1.6349	8.8127	4.6498	0.073
	+ SFT+TF	**0.408**	**0.643**	**0.545**	**1.610**	**0.276**	0.2383	0.3789	0.5273	5.8575	7.7830	5.0234	0.085
KIMODO	ExMoCap	0.990	0.998	0.540	1.527	–	–	–	–	–	–	–	0.071
	HumanML3D	–	–	–	–	–	0.4591	0.6649	0.7744	0.0010	9.3338	3.2309	–
	Baseline	0.223	-0.051	0.137	0.281	0.080	**0.3555**	**0.5547**	**0.6562**	2.0376	8.5198	**3.8190**	**0.025**
	+ DMP	0.208	0.053	0.374	1.086	0.144	0.1445	0.2773	0.3516	38.1156	5.7478	6.7370	0.069
	+ TF	0.208	0.039	0.287	0.801	0.081	0.2148	0.3516	0.4961	20.0688	6.8045	5.3506	0.036
	+ SFT	0.515	0.814	0.587	1.610	0.183	0.3320	0.5234	0.6289	**1.6934**	8.9660	3.9654	0.027
	+ SFT+TF	**0.531**	**0.831**	**0.729**	**2.015**	**0.306**	0.1992	0.3594	0.4648	18.9754	7.2581	5.5249	0.030



(a) Ablation on r and λ using Classifier-based exaggeration and R-Precision metrics.



(b) Ablation on r and λ using Classifier-free exaggeration and R-Precision metrics.

Figure 5: **Ablation study of LoRA rank and prior-preservation weight for MDM.** We evaluate the trade-off between exaggeration control and prompt adherence by varying the LoRA rank r (left) and prior-preservation weight λ (right) for MDM (Tevet et al., 2023). Higher ELRA@1 and Δ_{phys} indicate stronger exaggeration control, while higher R-precision indicates better prompt adherence. The pre-trained MDM baseline is denoted by a star.

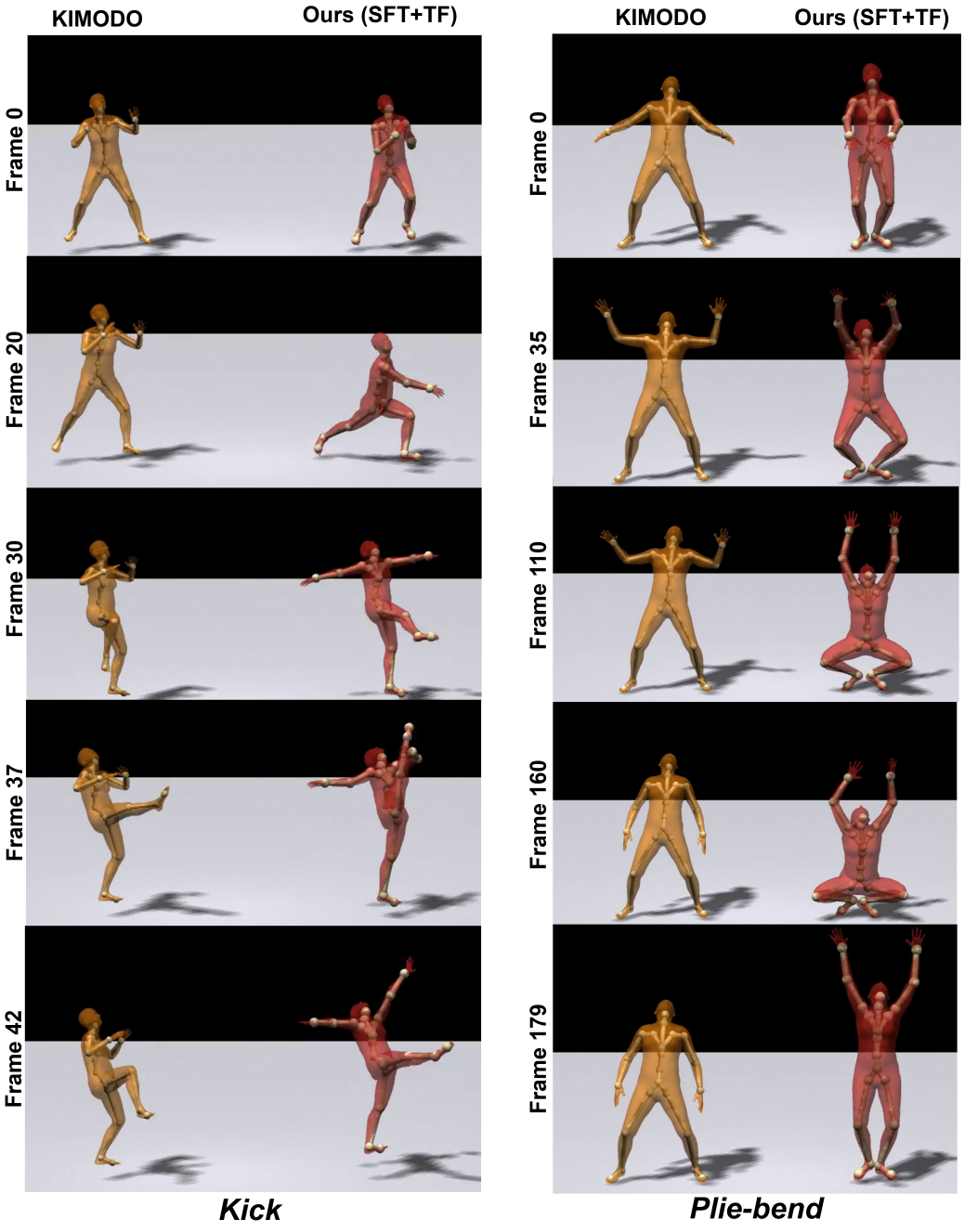


Figure 6: **Qualitative evaluation of our SFT+TF approach compared to KIMODO.** Selected frames from *Kick* and *Plie-bend* actions show that SFT+TF amplifies the characteristic poses and motion amplitudes of the original actions generated by KIMODO, producing more expressive exaggeration while preserving the underlying neutral action and its temporal progression.

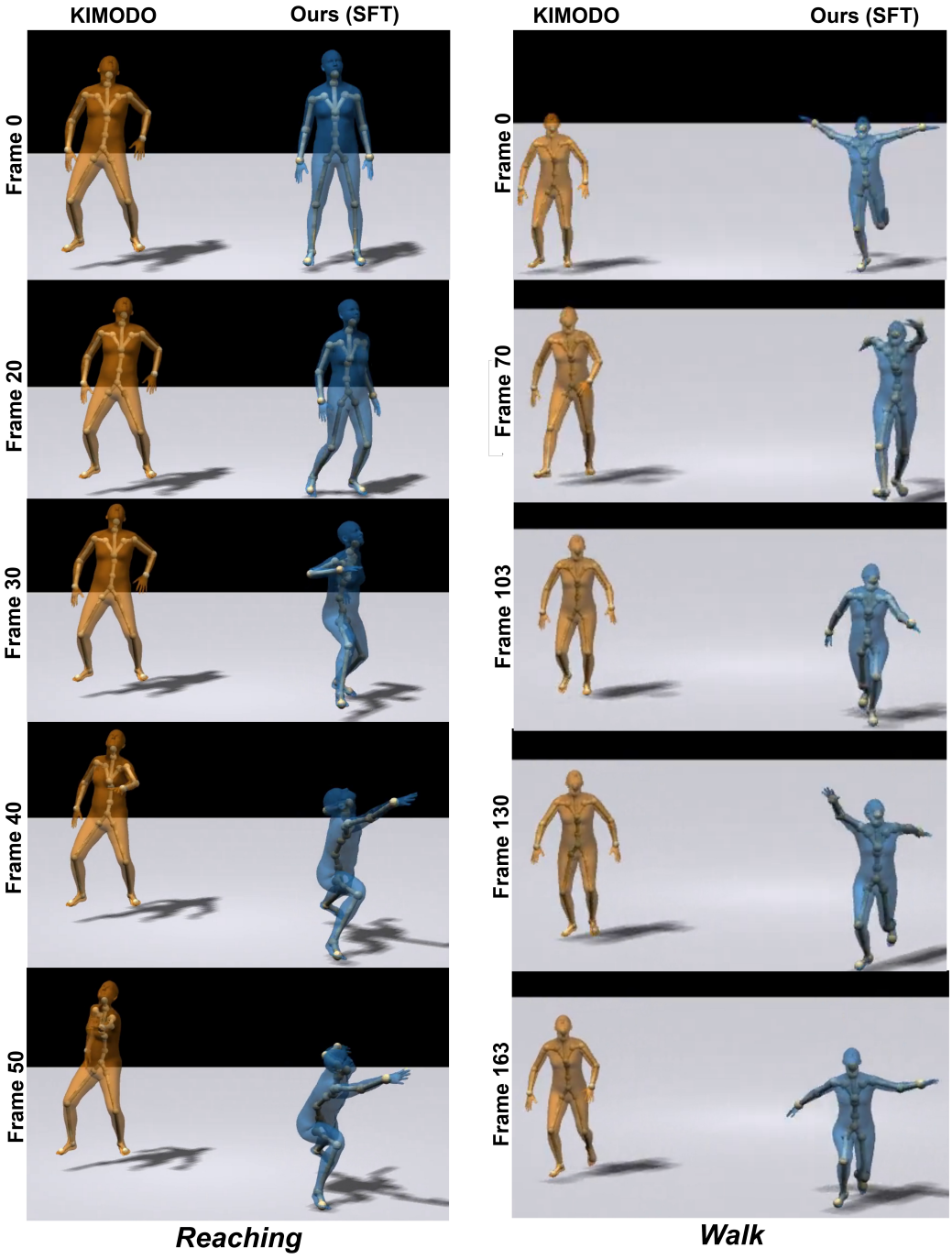


Figure 7: **Qualitative evaluation of our SFT approach compared to KIMODO.** Selected frames from *Reaching* and *Walk* actions show that SFT amplifies the characteristic poses and motion amplitudes of the original actions generated by KIMODO, producing more expressive exaggeration while preserving the underlying neutral action and its temporal progression.

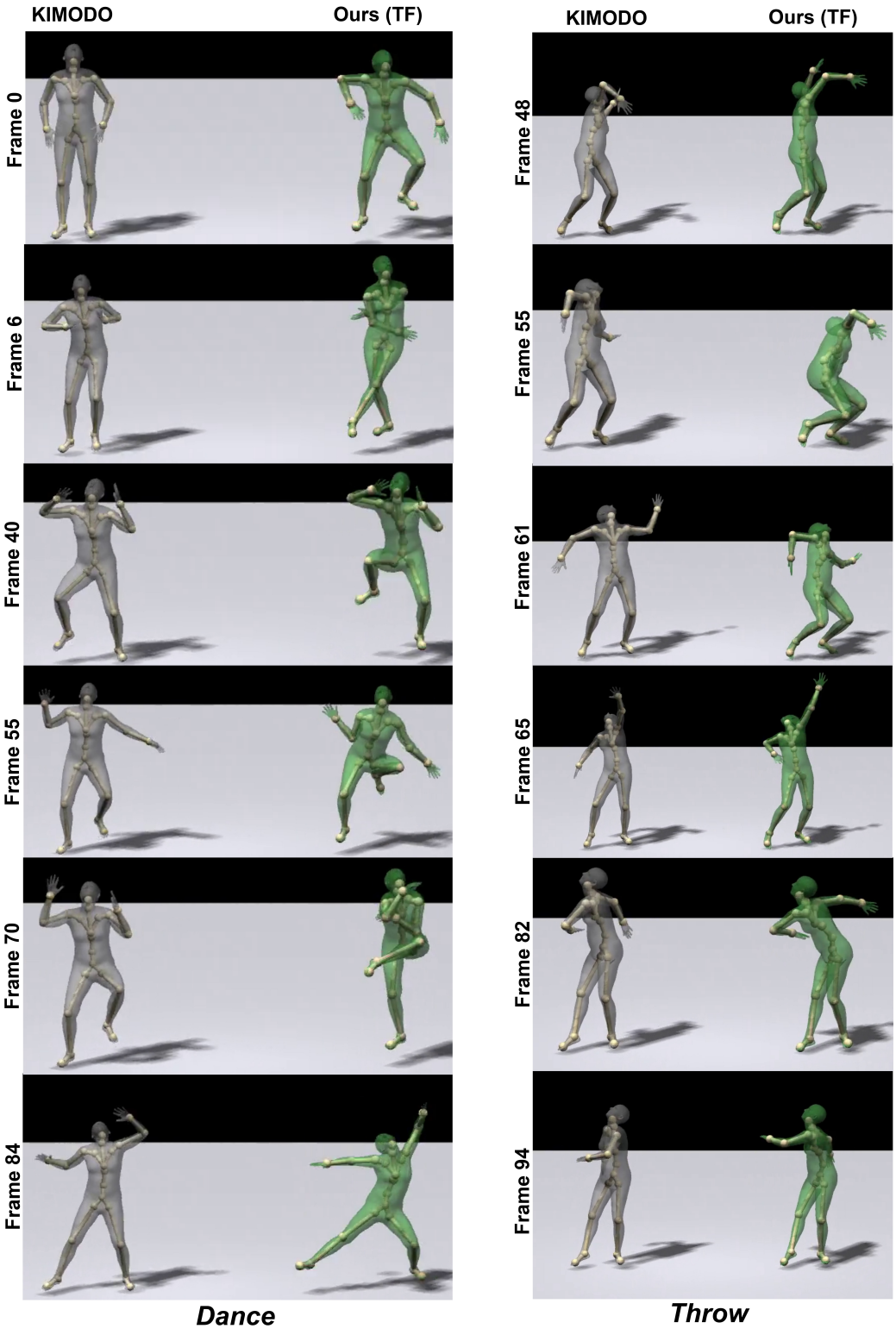


Figure 8: **Qualitative evaluation of our TF approach compared to KIMODO.** Selected frames from *Dance* and *Throw* actions show that TF amplifies the characteristic poses and motion amplitudes of the original actions generated by KIMODO, producing more expressive exaggeration while preserving the underlying neutral action and its temporal progression.

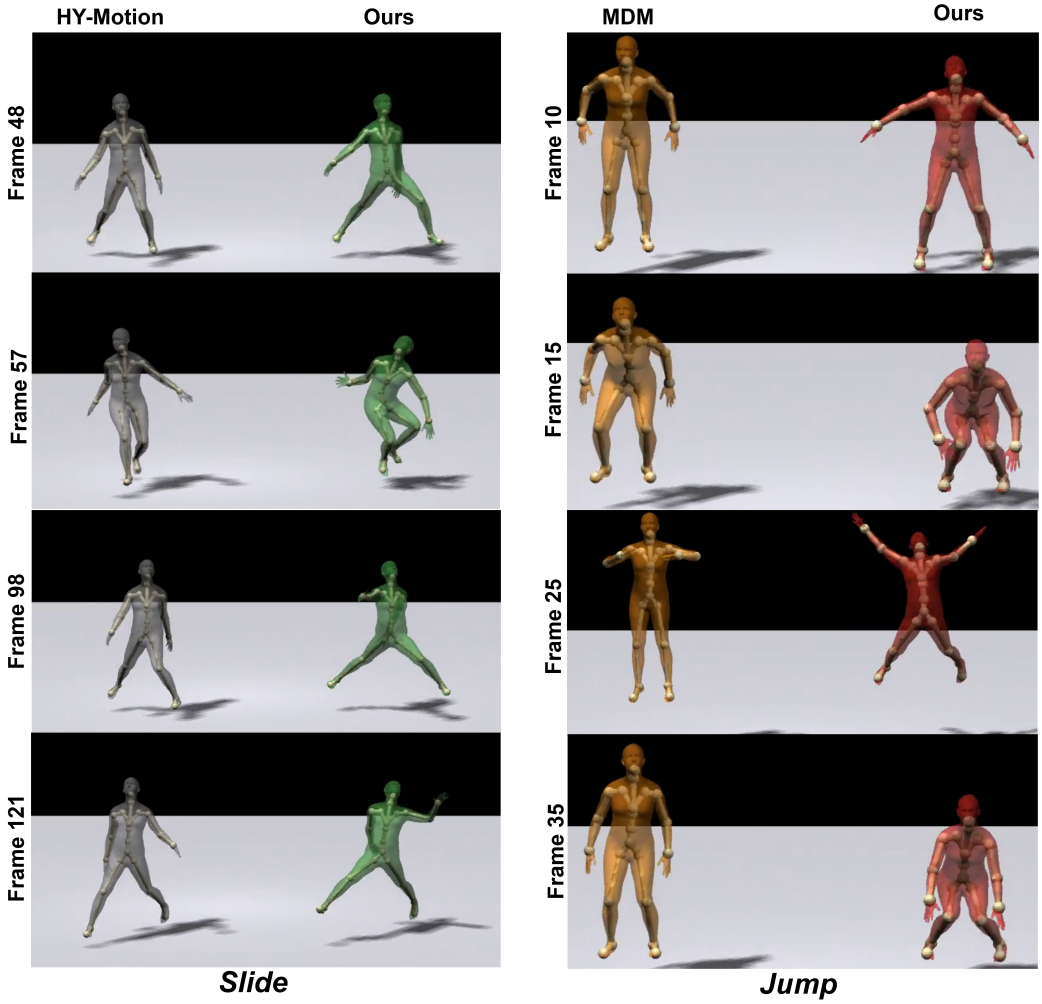


Figure 9: **More qualitative results on MDM and HY-Motion.** Selected frames from *Slide* and *Jump* actions compare the baseline models with our approach. Across both motion generation models, our method amplifies characteristic poses and motion amplitudes while preserving the underlying neutral action and temporal progression.

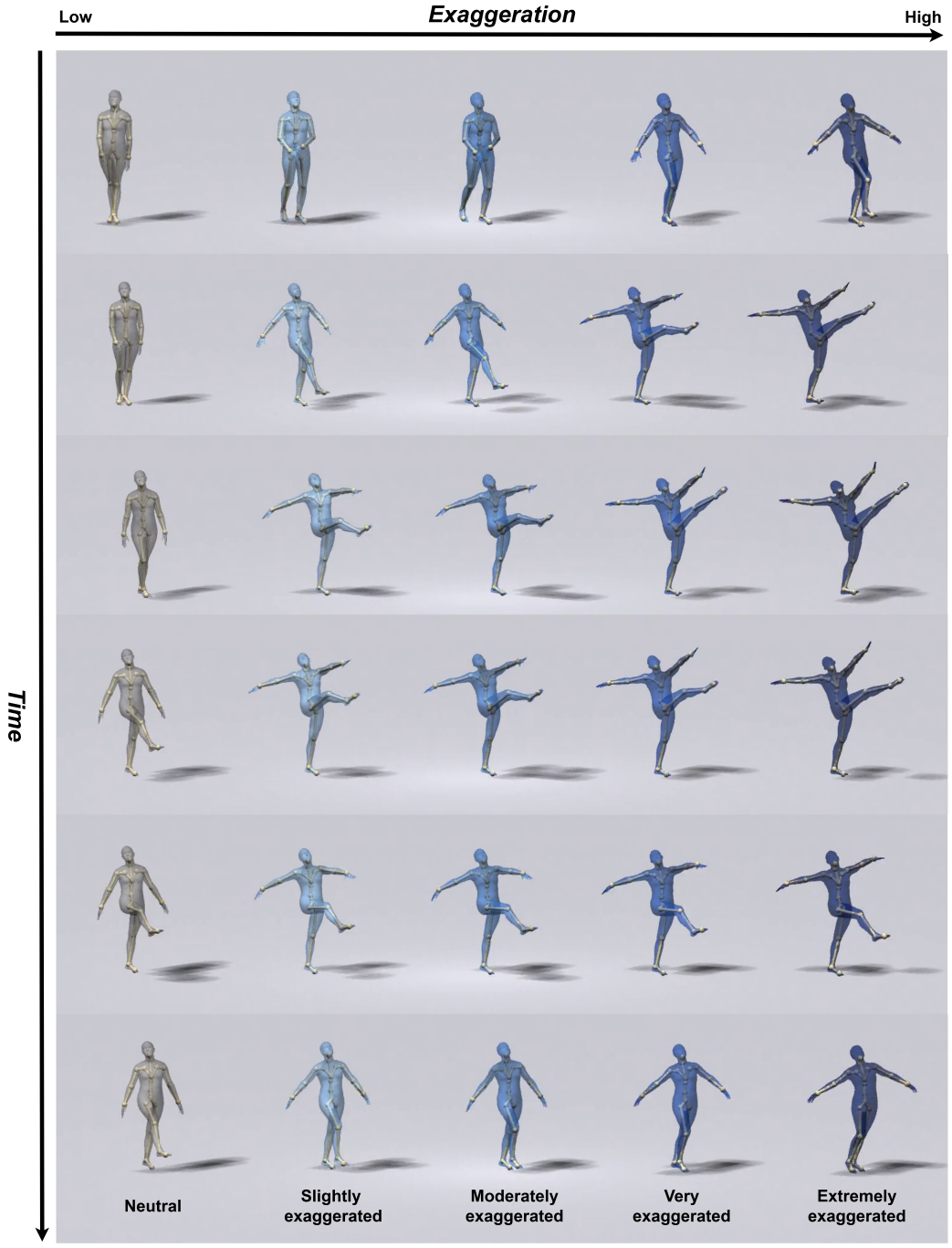


Figure 10: **Controllable exaggeration using our proposed supervised fine-tuning (SFT) approach.** A same kick motion is generated at progressively increasing exaggeration levels from left to right, with time progressing from top to bottom. Increasing exaggeration intensity through text prompt in the SFT approach produces progressively larger and more expressive poses while preserving the underlying neutral action.

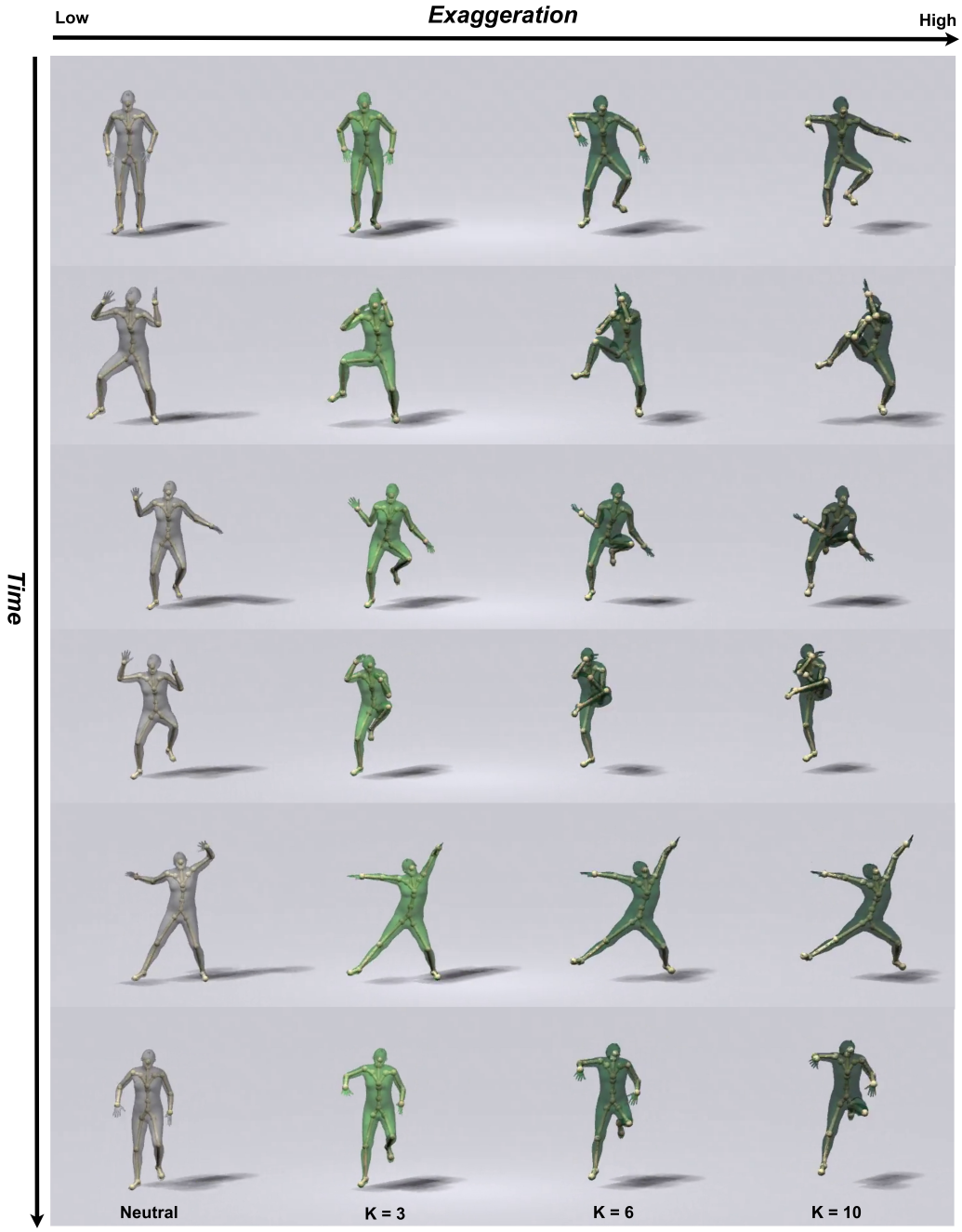


Figure 11: **Controllable exaggeration using our proposed training-free (TF) approach.** A same dance motion is generated at progressively increasing exaggeration levels from left to right, with time progressing from top to bottom. Increasing exaggeration intensity through the parameter k in the TF approach produces progressively larger and more expressive poses while preserving the underlying neutral action.

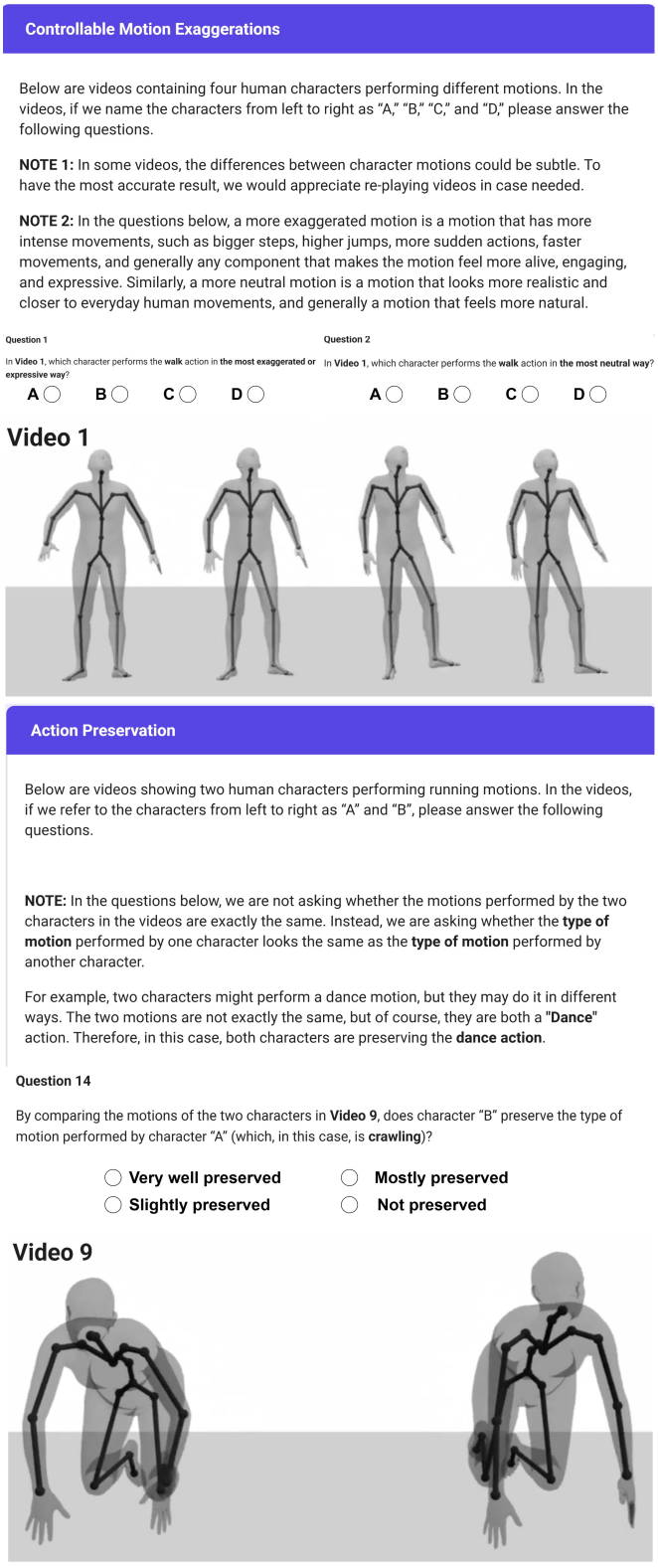


Figure 12: **Representative questions from our user study.** Top: for the controllable-exaggeration evaluation, participants viewed anonymized motions at different exaggeration levels and selected the most exaggerated and most neutral motions. Bottom: for the action-preservation evaluation, participants compared a neutral source motion with its exaggerated counterpart and rated how well the counterpart preserved the source action using a four-level response scale.