ProjFlow: Projection Sampling with Flow Matching for Zero-Shot Exact Spatial Motion Control

Akihisa Watanabe[1*] Qing Yu[2] Edgar Simo-Serra[1] Kent Fujiwara[2]

[1]Waseda University [2]LY Corporation

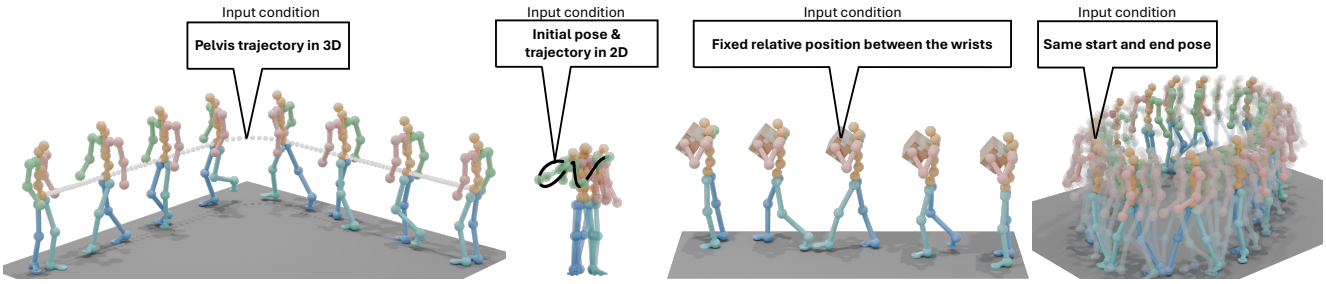


Figure 1. **ProjFlow** provides a unified, zero-shot framework for exact spatial motion control. The method handles diverse applications by formulating them as linear inverse problems. Examples of applications include (a) precisely following a specified joint's trajectory, (b) lifting 2D keypose and 2D trajectory inputs to a full 3D motion, (c) maintaining a fixed relative position between joints, and (d) generating seamlessly looped motion by matching start and end poses.

Abstract

Generating human motion with precise spatial control is a challenging problem. Existing approaches often require task-specific training or slow optimization, and enforcing hard constraints frequently disrupts motion naturalness. Building on the observation that many animation tasks can be formulated as a linear inverse problem, we introduce ***ProjFlow****, a training-free sampler that achieves zero-shot, exact satisfaction of linear spatial constraints while preserving motion realism. Our key advance is a novel kinematics-aware metric that encodes skeletal topology. This metric allows the sampler to enforce hard constraints by distributing corrections coherently across the entire skeleton, avoiding the unnatural artifacts of naive projection. Furthermore, for sparse inputs, such as filling in long gaps between a few keyframes, we introduce a time-varying formulation using pseudo-observations that fade during sampling. Extensive experiments on representative applications, motion inpainting, and 2D-to-3D lifting, demonstrate that ProjFlow achieves exact constraint satisfaction and matches or improves realism over zero-shot baselines, while remaining competitive with training-based controllers.*

*Work done during an internship at LY Corporation.

1. Introduction

An open challenge in character animation is spatial motion control, which involves generating realistic full-body motion that conforms to user-defined spatial cues. These cues can include trajectories, target poses, or specific joint locations. Solving this task would allow 3D animators to work with precise and interactive control, immediately obtaining desired motions that remain natural and diverse [1, 54].

Users typically specify constraints for only a subset of the body, such as the trajectory of a single hand or foot. This makes the spatial motion control problem ill-posed, with many motions satisfying these sparse constraints. An intuitive approach to resolve this ambiguity is to favor motions with high likelihood under a pretrained motion prior, selecting the most natural result from all valid options.

Building on this idea, dominant approaches steer pretrained diffusion models to satisfy user-defined spatial constraints. However, existing methods suffer from significant limitations. They often require task-specific training for conditioning branches [9, 39, 45, 61], or they rely on slow, inference-time optimization [21, 44, 45, 47], which reduces interactivity and can get stuck in local minima. Fundamentally, these approaches treat constraints as soft objectives rather than hard rules. As a result, exact satisfaction is not guaranteed, and residual violations persist. What is missing

is a sampler that can (i) enforce hard equality constraints exactly, (ii) operate zero-shot without task-specific retraining, and (iii) require no inner-loop optimization at inference time, all while preserving the pretrained motion prior.

In this paper, we present **ProjFlow**, **Proj**ection Sampling with **Flow** Matching for zero-shot exact spatial motion control. We begin with the observation that a wide range of motion control and editing tasks can be formulated as linear inverse problems. These tasks include trajectory following, keyframing, camera or root path control, and partial-body editing. ProjFlow addresses these problems by projecting the predicted clean motion at every denoising step onto the set of motions that satisfy the given constraints. This projection introduces the smallest necessary adjustment, measured under a newly designed *kinematics-aware metric* that reflects skeletal topology. Rather than measuring the distance in Euclidean space, this metric ensures that updates propagate coherently along the kinematic tree, avoiding unnatural and isolated joint movements. Hard constraints are satisfied exactly, while uncertain or partial measurements are weighted according to their confidence. The projected update is then combined with a flow-matching recomposition step, preserving the pretrained motion prior without any task-specific retraining or inner-loop optimization.

We evaluate the versatility of the ProjFlow framework through two representative applications in spatial motion control. The first application is motion inpainting, where segments of a motion sequence are entirely missing. This task requires the model to infer plausible intermediate frames from sparse temporal observations. Instead of treating the unobserved frames as blanks, ProjFlow introduces pseudo-observations around known frames and gradually adjusts their influence during sampling, enabling coherent zero-shot completion even across long temporal gaps.

The second application is 2D-to-3D motion reconstruction, where the input consists of 2D keypoints and their trajectories over time. The goal is to recover the underlying 3D motion that projects onto the observed 2D data. ProjFlow enforces linear measurement constraints derived from the camera model as hard equalities at each step. This yields accurate 3D reconstructions with zero reprojection error and natural motion. Our experiments on these applications show ProjFlow matches the accuracy of training-based methods without any retraining or inner-loop optimization. These results demonstrate the versatility of our framework, which can also be applied to the other tasks illustrated in Fig. 1.

In summary, our contributions are as follows:

- **Unified linear inverse formulation and projection sampler as its solver.** We cast motion control and editing as linear inverse problems and propose a projection-based flow-matching sampler that enforces constraints exactly without retraining or inner-loop optimization.
- **Kinematics-aware projection geometry.** We introduce a metric that encodes skeletal structure, providing a principled geometry that distributes corrections coherently and improves realism and stability.
- **Empirical parity on inpainting and 2D-to-3D with exact constraints.** Through experiments on motion inpainting and 2D-to-3D reconstruction, we show that ProjFlow matches the performance of training-based models while satisfying the specified constraints exactly up to numerical precision, all in a zero-shot, no inner loop setting.

2. Related Work

2.1. Human Motion Generation

Recent advances in image generation indicate a transition from denoising diffusion probabilistic models and score-based SDEs to flow matching models that learn velocity fields using rectified-flow objectives, scaling well with Transformer architectures [13, 17, 33, 34, 36, 53]. Progress in text-conditioned human motion generation has followed the same arc. Early state-of-the-art systems were diffusion-based [8, 56, 64, 65], while more recent work adopts flow-matching formulations [4, 18].

Alongside advances in generative methodology, motion representation has also evolved. HumanML3D [16] popularized a kinematic, relative, and partly redundant feature representation still adopted by many controllers [9, 16, 20, 61]. Evidence now shows that generating absolute joint coordinates in world space with a rectified-flow objective is effective and beneficial for controllability and scalability [39, 40]. These trends motivate our choice of a flow-matching sampler operating directly in world coordinates.

2.2. Spatially Controlled Motion Generation

While text prompts are effective for controlling high-level motion semantics, many practical applications require more precise spatial control. Synthesizing motion from a wider range of external control signals, often in combination with text prompts, has been widely explored. Examples include authoring from storyboard sketches [67] and multi-track timeline authoring [43]. Other research streams focus on multi-objective control for characters and robots [3, 49], music-conditioned choreography [25, 26, 28–30, 57], or generating motions involving inter-human [14, 32, 41, 55] and human-object interactions [5, 10, 24, 27]. Control signals can also include sparse tracking inputs [12], scene affordances [19, 60], programmable objectives [35], style specifications [66], or goal-directed targets [11].

A key question is how to effectively integrate these spatial signals into text-to-motion generators to enforce precise accuracy. Prior work has taken several routes to tackle this. One approach involves fine-tuning diffusion priors with end-effector supervision [50] or training models for in-betweening from dense or sparse keyframes [7]. Another line of work applies guidance during sampling, steer-

ing the generation towards root or waypoint trajectories [20, 46]. More recently, joint-wise conditioning has been achieved using ControlNet-style branches or latent controllers [9, 61, 63]. Others perform inference-time optimization of the initial noise or logits to minimize differentiable objectives [21, 45], or use factorization and controller mixtures for fine-grained control [31, 58]. Across these routes, constraints are injected as differentiable penalties or guidance terms rather than enforced as hard feasibility constraints. Consequently, exact feasibility is not guaranteed, and methods often require task-specific conditioning or iterative inner-loop optimization during inference.

2.3. Inverse Problems with Image Generation

Pre-trained diffusion priors have enabled strong zero-shot solvers for linear inverse problems. Two influential views have emerged. The first is likelihood guidance along the sampling path [6, 22]. The second is *projection* that freezes range-space and refines only the null-space (DDNM) [59], with extensions such as pseudoinverse guidance [52]. To leverage large latent generative models, latent diffusion model-based variants inject data consistency in latent space [48, 51, 62]. Recently, these ideas have been extended to flow models. FlowChef and PnP-Flow steer rectified-flow fields or plug a learned denoiser into a flow solver [38, 42], but do not cast inverse solving as closed-form posterior steps on the flow path.

ProjFlow adapts data consistency updates to the flow matching regime, and the framework generalizes prior posterior projection samplers in two key ways. First, it replaces the common Euclidean geometry of image methods with a kinematics-aware metric that distributes corrections coherently along the skeleton, which better supports structured data such as human motion. Second, the framework introduces time-scheduled pseudo-observations that densify guidance in unobserved regions and then fade as sampling proceeds, improving on prior approaches that treat missing regions as simple blanks. Finally, ProjFlow recovers DDNM in the Euclidean noiseless deterministic limit while extending support to structured metrics, noisy measurements, and time-varying operators.

3. Preliminaries

3.1. Motion Representation

We represent a clean motion sequence of length N with J joints in absolute world coordinates as a tensor $\boldsymbol{x} \in \mathbb{R}^{N\times J\times 3}$. For brevity, we also use $\boldsymbol{x}$ to denote its vectorization $\boldsymbol{x} \in \mathbb{R}^d$ with $d = 3JN$. Unless stated otherwise, we assume a frame-major order. Each vector element $i \in \{1, \dots, d\}$ corresponds to a unique frame–joint–spatial-channel triple (n_i, j_i, c_i), where $n_i \in \{1, \dots, N\}$, $j_i \in \{1, \dots, J\}$, and $c_i \in \{x, y, z\}$.

3.2. Flow Matching

The core idea of flow-based generative models [2, 33, 36] is to learn a time-dependent vector field $v_\theta(\boldsymbol{x}, t)$ that transports samples from a simple prior distribution p_0 to a complex target data distribution q.

Let $\psi_t : \mathbb{R}^d \to \mathbb{R}^d$ denote the flow map induced by this vector field. The flow map is defined as the unique solution to the Ordinary Differential Equation (ODE)

$$\frac{d\psi_t(\boldsymbol{x}_0)}{dt} = v_\theta\big(\psi_t(\boldsymbol{x}_0), t\big), \qquad \psi_0(\boldsymbol{x}_0) = \boldsymbol{x}_0, \tag{1}$$

where $\boldsymbol{x}_0$ is the initial condition.

In this study, we adopt the Rectified Flow formulation [34, 36], which defines a straight-line path between a noise sample $\boldsymbol{x}_0$ and a data sample $\boldsymbol{x}_1$:

$$\boldsymbol{x}_t = (1-t)\,\boldsymbol{x}_0 + t\,\boldsymbol{x}_1, \qquad t \in [0, 1]. \tag{2}$$

Along this path, the ideal velocity is constant and equal to $\boldsymbol{x}_1 - \boldsymbol{x}_0$. The network v_θ is trained to approximate the conditional expectation of this velocity given $(\boldsymbol{x}_t, t)$ by minimizing the conditional flow-matching loss

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{\substack{t\sim\mathcal{U}(0,1)\\ \boldsymbol{x}_0\sim p_0\\ \boldsymbol{x}_1\sim q}}\Big[\,\big\|v_\theta(\boldsymbol{x}_t, t) - (\boldsymbol{x}_1 - \boldsymbol{x}_0)\big\|_2^2\Big], \tag{3}$$

where $\boldsymbol{x}_t$ is given by equation 2. Sampling is then performed by drawing $\boldsymbol{x}_0 \sim p_0$ and numerically integrating the ODE in equation 1 from $t = 0$ to $t = 1$ to obtain $\boldsymbol{x}_1 = \psi_1(\boldsymbol{x}_0)$.

This formulation provides a continuous and differentiable generative path between the prior and data distributions, which later facilitates direct constraint enforcement in our projection-based framework.

4. Method

In this section, we first formulate spatial control as a unified linear inverse problem (Sec. 4.1). We then introduce ProjFlow, our kinematics-aware projection sampler (Sec. 4.2), and demonstrate its use in representative applications (Sec. 4.3).

4.1. Spatial Motion Control as a Linear Inverse Problem

We unify all user-specified constraints into a single linear observation model

$$\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \Sigma), \tag{4}$$

where $\boldsymbol{y} \in \mathbb{R}^m$ is the vector of user-specified observed measurements, $A : \mathbb{R}^d \to \mathbb{R}^m$ is a known linear operator, and $\Sigma \succeq \mathbf{0}$ is an observation noise covariance. Hard constraints are recovered as the limiting case where the corresponding rows of Σ tend to zero variance.

Our objective is to generate a motion $\hat{\boldsymbol{x}}$ that is consistent

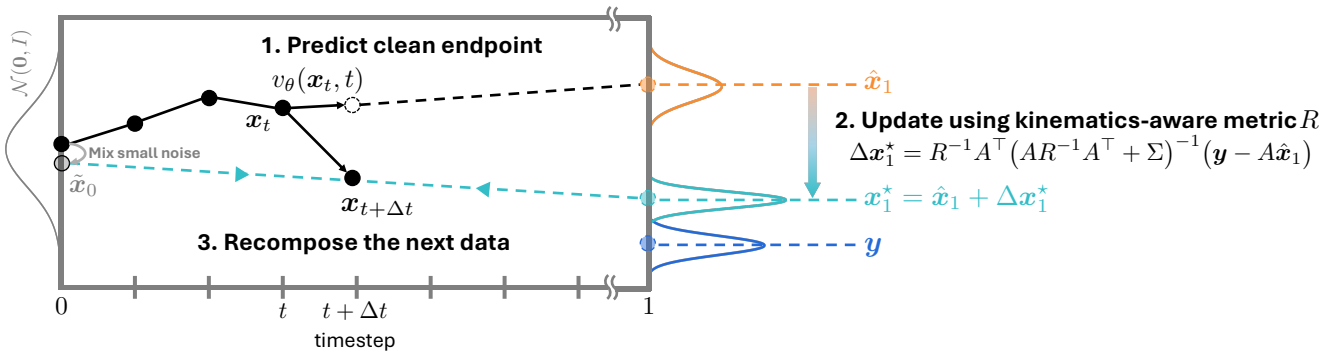


Figure 2. **Overview of the Projection Sampling Step.** At each timestep t: (1) predict the clean endpoint $\hat{\boldsymbol{x}}_1$ from $\boldsymbol{x}_t$ using the learned velocity $v_\theta(\boldsymbol{x}_t, t)$; (2) enforce the linear–Gaussian measurements $\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{\epsilon}$ by computing a correction $\Delta\boldsymbol{x}_1^\star$ that projects $\hat{\boldsymbol{x}}_1$ to the measurement set under the *kinematics-aware* metric R. This metric encodes skeletal topology and spreads updates coherently along the kinematic tree. The measurement covariance Σ modulates the pull toward the observations; smaller values yield stronger attraction and recover hard constraints as $\Sigma \to 0$. (3) Finally, stochastically recompose the corrected endpoint to obtain the next state $\boldsymbol{x}_{t+\Delta t}$.

with the observation model equation 4 while maintaining the realism encoded in the pretrained motion prior.

4.2. Projection Sampling with Flow Matching

Given the intermediate state $\boldsymbol{x}_t$ and the predicted velocity $v_\theta(\boldsymbol{x}_t, t)$, as shown in Fig. 2, the corresponding clean endpoints can be obtained by Tweedie's formula [23]

$$\hat{\boldsymbol{x}}_1 = \mathbb{E}[\boldsymbol{x}_1|\boldsymbol{x}_t] = \boldsymbol{x}_t + (1-t)v_\theta(\boldsymbol{x}_t, t). \tag{5}$$

We seek the smallest clean-endpoint correction $\Delta\boldsymbol{x}_1$ (in the metric $R \succ 0$) by solving the problem

$$\min_{\Delta\boldsymbol{x}_1} \frac{1}{2}\|\Delta\boldsymbol{x}_1\|_R^2 + \frac{1}{2}\|\boldsymbol{y} - A(\hat{\boldsymbol{x}}_1 + \Delta\boldsymbol{x}_1)\|_{\Sigma^{-1}}^2. \tag{6}$$

This convex quadratic problem has a unique closed-form solution $\Delta\boldsymbol{x}_1^\star$ given by

$$\Delta\boldsymbol{x}_1^\star = R^{-1}A^\top\left(AR^{-1}A^\top + \Sigma\right)^{-1}\left(\boldsymbol{y} - A\hat{\boldsymbol{x}}_1\right). \tag{7}$$

Applying this to $\hat{\boldsymbol{x}}_1$ yields the corrected clean endpoint

$$\hat{\boldsymbol{x}}_1^\star = \hat{\boldsymbol{x}}_1 + \Delta\boldsymbol{x}_1^\star. \tag{8}$$

We then compute the next state $\boldsymbol{x}_{t+\Delta t}$ by adapting the stochastic recomposition step from the FlowDPS sampler [23]. This step combines our corrected clean endpoint $\hat{\boldsymbol{x}}_1^\star$ with a mixed version of the *original* noise $\boldsymbol{x}_0$:

$$\tilde{\boldsymbol{x}}_0 = \sqrt{1-\eta_t}\boldsymbol{x}_0 + \sqrt{\eta_t}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, I) \tag{9}$$

$$\boldsymbol{x}_{t+\Delta t} = \alpha_{t+\Delta t}\hat{\boldsymbol{x}}_1^\star + \sigma_{t+\Delta t}\tilde{\boldsymbol{x}}_0, \tag{10}$$

where η_t is a noise-mixing parameter, and the path coefficients are defined as $\alpha_{t+\Delta t} = t + \Delta t$ and $\sigma_{t+\Delta t} = 1 - (t + \Delta t)$.

Kinematics-aware Metric The choice of metric R determines how we measure the size of a correction $\Delta\boldsymbol{x}_1$ in the clean motion space. With the Euclidean metric ($R = I$), all coordinates are weighted equally, so slight changes to a few joints may appear "small" in terms of ℓ_2 norm even if it breaks kinematic coherence. We instead define smallness by coherence along the kinematic tree. The full metric R for a motion $\boldsymbol{x} \in \mathbb{R}^d$ is defined as

$$R = w_{\text{kin}}(I_3 \otimes I_N \otimes L_{\text{kin}}) + \lambda I_d, \tag{11}$$

where $L_{\text{kin}} \in \mathbb{R}^{J\times J}$ is the standard unnormalized graph Laplacian of the skeletal topology. It is constructed from the skeleton's adjacency matrix A_{kin} (where $(A_{\text{kin}})_{j_1 j_2} = 1$ if joint j_1 and j_2 are connected) as $L_{\text{kin}} = D_{\text{kin}} - A_{\text{kin}}$, with the diagonal degree matrix $D_{\text{kin}} = \text{diag}(A_{\text{kin}}\mathbf{1})$. I_k is the $k \times k$ identity matrix, w_{kin} is a scalar weight for the kinematic term, and $\lambda > 0$ is a weight for the identity term, which ensures R is strictly positive definite and invertible. This metric is applied independently to each of the $x, y,$ and z spatial dimensions via the I_3 term.

This metric makes the intended measurement of "small" explicit (i) discrepancies across *adjacent joints* are strongly penalized by the kinematic term $w_{\text{kin}}L_{\text{kin}}$, while joints that are not directly connected in the kinematic tree incur little coupling, reflecting the skeletal topology. (ii) The identity term λI adds a baseline ℓ_2 penalty to directions, which are per-frame global translations that are not penalized by the kinematic component. This penalty regularizes these otherwise unconstrained modes and ensures that the full metric R is strictly positive definite.

4.3. Spatial Control with ProjFlow

We illustrate ProjFlow in practice through two representative spatial control applications: motion inpainting and 2D-to-3D lifting. Other extensions, such as motion loop closure and relative body part control shown in Fig. 1, are formulated in the supplementary material.

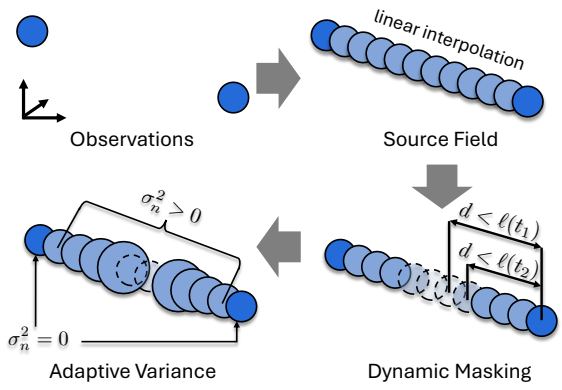


Figure 3. **Pseudo-observations for motion inpainting.** Sparse observations are interpolated to guide intermediate frames. This guidance is controlled by two mechanisms: Dynamic Masking activates a time-scheduled neighborhood, and Adaptive Variance treats original observations as hard constraints and the interpolated guides as soft constraints.

4.3.1. Application I: Motion Inpainting via Masked Pseudo-Observations

Plain Masking. We cast inpainting as recovering the full motion vector $\boldsymbol{x} \in \mathbb{R}^d$ from sparse hard observations, such as keyframe joint locations provided by users. Let $M_{\text{obs}} \in \{0,1\}^{d\times d}$ be a diagonal mask selecting observed coordinates, and $\boldsymbol{y}_{\text{obs}} \in \mathbb{R}^d$ store their values (zeros elsewhere). The hard-constraint model is

$$\boldsymbol{y}_{\text{obs}} \;=\; M_{\text{obs}}\boldsymbol{x}. \tag{12}$$

Time-varying Pseudo-observations. When these hard observations are sparse, the model provides insufficient guidance. We therefore introduce "soft" pseudo-observations $\boldsymbol{y}_{\text{src}}$, created via per-joint linear interpolation, to provide denser guidance. However, these pseudo-observations from linear interpolation are not always reliable. We want the variance to be high (i.e., trust is low) in two cases (i) As sampling progresses ($t \to 1$), we trust the model's own prediction $\hat{\boldsymbol{x}}_1$ more. (ii) Where motion curvature is high, linear interpolation is a poor estimate.

We combine these soft guides with the hard observations $\boldsymbol{y}_{\text{obs}}$ to formulate a time-varying linear inverse problem at each sampling step t

$$\boldsymbol{y}^{(t)} \;=\; M^{(t)}\boldsymbol{x} + \boldsymbol{\epsilon}^{(t)}, \qquad \boldsymbol{\epsilon}^{(t)} \sim \mathcal{N}\big(0, \Sigma^{(t)}\big), \tag{13}$$

where $M_{\text{aug}}^{(t)}$ is a diagonal matrix activating pseudo-observations within a temporal neighbourhood of hard constraints, but explicitly excluding the hard constraints themselves. The combined mask is the union of these disjoint sets, $M^{(t)} = M_{\text{obs}} + M_{\text{aug}}^{(t)}$. The target observation is $\boldsymbol{y}^{(t)} = \boldsymbol{y}_{\text{obs}} + M_{\text{aug}}^{(t)}\,\boldsymbol{y}_{\text{src}}$. The diagonal covariance $\Sigma^{(t)} = \text{diag}(\sigma_1^2(t), \ldots, \sigma_d^2(t))$ assigns an adaptive, non-zero variance $\sigma_i^2(t) > 0$ to the active pseudo-observations based on their reliability. The actual observations are treated as exact linear equalities.

Dynamic Masking. The temporal neighbourhood of pseudo-observations (Fig.3, Dynamic Masking) shrinks linearly in time. This mechanism gradually phases out the soft pseudo-observations, leaving only the hard constraints active as $t \to 1$. We define this shrinking radius $\ell(t)$ as

$$\ell(t) \;=\; (1-t)\,\ell_{\max} + t\,\ell_{\min}. \tag{14}$$

A frame's pseudo-observations are activated only if the temporal distance to its nearest hard observation is less than this radius $\ell(t)$.

Adaptive Variance. We control the reliability of the pseudo-observations by setting their variance $\sigma_i^2(t)$ (Fig.3, Adaptive Variance). We model the trust level with a frame-wise score $\tilde{\pi}_n^{(t)}$

$$\tilde{\pi}_n^{(t)} \;=\; \tau(t)\,\frac{c_0}{1 + \lambda_s\,(s_n(\hat{\boldsymbol{x}}_1)/s_{\text{med}})^p} \tag{15}$$

where c_0, λ_s, and p are hyperparameters controlling the adaptive strength. This score combines a global time-decay term,

$$\tau(t) \;=\; \tau_{\min} + (1-\tau_{\min})(1-t), \tag{16}$$

where $\tau_{\min}$ is a hyperparameter, with a local curvature penalty $s_n(\hat{\boldsymbol{x}}_1)$, defined as

$$s_n(\hat{\boldsymbol{x}}_1) \;=\; \|(\hat{\boldsymbol{x}}_1)_{n+1} - 2(\hat{\boldsymbol{x}}_1)_n + (\hat{\boldsymbol{x}}_1)_{n-1}\|_R. \tag{17}$$

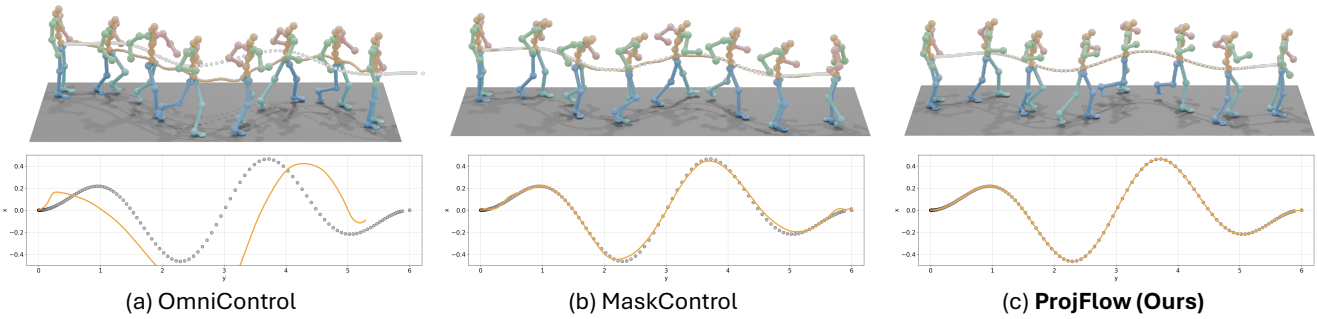
Here, s_{med} is the median curvature $s_n(\hat{\boldsymbol{x}}_1)$ across the sequence, used for robust normalization. As time t increases or curvature s_n increases, the trust score $\tilde{\pi}_n^{(t)}$ decreases. We clip this score to get a frame-level base target $\pi_n^{(t)} = \text{clip}\big(\tilde{\pi}_n^{(t)},\, \pi_{\min},\, \pi_{\max}\big)$. This base score is then modulated per-joint based on the properties of the kinematic metric to yield the final per-element score π_i. This π_i is used to compute the variance $\sigma_i^2(t)$ for the active pseudo-observation via the relation $\pi_i = r_i/(r_i + \sigma_i^2(t))$. Solving for the variance gives

$$\sigma_i^2(t) \;=\; r_i\,\frac{1-\pi_i}{\pi_i} \tag{18}$$

where $r_i = [\text{diag}(R^{-1})]_i$ is the i-th diagonal element of the inverse kinematic metric. Hard observations always maintain zero variance ($\sigma_i^2 = 0$).

4.3.2. Application II: 2D-to-3D Lifting via Linear Projection Measurements

The 2D-to-3D motion lifting task can also be expressed as a linear inverse problem. In this setting, we assume noise-free hard constraints, so the model simplifies to $\boldsymbol{y} = A\,\boldsymbol{x}$. The operator A maps the vectorized 3D motion sequence $\boldsymbol{x}$ to stacked 2D joint coordinates. This operator is constructed



(a) OmniControl (b) MaskControl (c) **ProjFlow (Ours)**

Figure 4. **Text-conditioned pelvis-trajectory control.** Given the prompt "*a person runs forward in an S-shaped path*" and a pelvis control signal, we compare OmniControl [61], MaskControl [45], and ProjFlow (ours). The rendered motions and the trajectory plots both visualize the generated pelvis trajectory (orange) overlaid on the target control signal (gray dotted line).

in two steps. First, we define a full projection operator A_{full} that maps all 3D joints at all frames to 2D. It does this by stacking the standard linear orthographic projection,

$$\boldsymbol{y}_{n,j} = sPR_{\text{cam}}\boldsymbol{x}_{n,j}, \tag{19}$$

for every frame n and joint j, where s is a fixed scale factor, $P = [\,1\ 0\ 0;\ 0\ 1\ 0\,]$ is the orthographic projection matrix, and $R_{\text{cam}} \in \mathrm{SO}(3)$ is the camera rotation. Both s and R_{cam} are assumed to be known for each sequence.

Second, we define a binary selection operator M that filters the rows of A_{full} to match the user's specific inputs (e.g., all joints at frame 0 and a subset of joints for $n > 0$). M is constructed to select only these corresponding rows. The final measurement operator A is therefore defined as

$$A = M\, A_{\text{full}}. \tag{20}$$

5. Experiments

In this section, we evaluate the performance of ProjFlow, comparing it to previous task-specific/zero-shot methods.

5.1. Experimental Setup

Datasets. We experiment on the popular HumanML3D [16] dataset which contains 14,646 text-annotated human motion sequences from AMASS [37] and HumanAct12 [15] datasets.

Evaluation Protocol. We adopt the pretrained ACMDM-S-PS22 [39] as our base flow-matching model for all experiments and primarily follow the protocol of Meng et al. [40]. For spatial control experiments, we follow the OmniControl [61] evaluation protocol, which varies the density of control signals across five settings (1, 2, 5, 49, and 196 keyframes), and report the mean of each control metric across these densities to assess robustness to sparsity.

For the 2D-to-3D task, we follow the Sketch2Anim [67] protocol, which defines camera parameters including pitch $\in [0°, 30°]$, yaw $\in [-45°, 45°]$, roll $= 0°$, and $s \in [0.8, 1.2]$. We evaluate under this known orthographic camera at inference time.

Evaluation Metrics. To assess generation quality and text alignment, we report *FID* for distribution similarity, *R-Precision (Top-1/2/3)* and *Matching Score* for semantic retrieval accuracy between motion and text embeddings, *Diversity* for motion diversity. For spatial control tasks, we evaluate accuracy using *Trajectory Error*, *Location Error*, and *Average Error*, which measure deviations from target keyframes at trajectory, keyframe, and mean distance levels, respectively. Physical plausibility is assessed via the *Foot Skating Ratio*.

For the 2D-to-3D reconstruction task, in addition to the above metrics, we report *MPJPE-2D* and *Avg. Err.-2D*. These metrics evaluate constraint satisfaction by projecting the generated 3D motion back into 2D and quantifying the mean error against the target 2D joint coordinates, following the protocol of Sketch2Anim [67].

5.2. Results

5.2.1. Motion Inpainting with Trajectory Control

Quantitative Performance. ProjFlow is the only zero-shot method that achieves *exact* constraint satisfaction (0.0000 on trajectory/location/average errors) while also attaining the best realism among zero-shot baselines. As shown in Table 1, its FID is lower than DNO(ACMDM-S-PS22+DNO) [21] for both pelvis control and all joints, which indicates that ProjFlow can eliminate the small residual violations that remain for guidance/noise-optimization methods.

Compared to models that require additional training, as shown in Table 1, ProjFlow stays in a similar realism band while remaining training-free and achieving *exact* constraint satisfaction. For example, MaskControl [45] reaches a lower FID but still leaves a non-zero average error (0.0093), whereas ProjFlow maintains all control errors at 0.0000. The same tendency is observed in other training-

Table 1. **Quantitative text-conditioned motion generation with spatial control signals and upper-body editing on HumanML3D[16].** In the first section, methods are trained and evaluated solely on pelvis controls. In the middle section, methods are trained on all joints and evaluated separately on each controlled joint. Only average results are reported for brevity. We include details in the supplementary material. The last section presents upper-body editing results. **bold** face / underline indicates the best/2nd results.

Controlling Joint	Methods	Zero-shot?	FID↓	R-Precision Top 3	Diversity→	Foot Skating Ratio.↓	Traj. err.↓	Loc. err.↓	Avg. err.↓
	GT	-	0.000	0.795	10.455	-	0.000	0.000	0.000
Pelvis	MDM [56]	✓	1.792	0.673	9.131	0.1019	0.4022	0.3076	0.5959
	PriorMDM [50]	✗	0.393	0.707	9.847	0.0897	0.3457	0.2132	0.4417
	GMD [20]	✓	0.238	0.763	10.011	0.1009	0.0931	0.0321	0.1439
	OmniControl [61]	✗	0.081	0.789	10.323	0.0547	0.0387	0.0096	0.0338
	MotionLCM V2+CtrlNet [9]	✗	3.978	0.738	9.249	0.0901	0.1080	0.0581	0.1386
	MaskControl [45]	✗	**0.066**	0.799	**10.474**	**0.0543**	**0.0000**	**0.0000**	0.0093
	ACMDM-S-PS22+CtrlNet [39]	✗	0.067	**0.805**	10.481	0.0591	0.0075	0.0010	0.0100
	ACMDM-S-PS22+DNO [21]	✓	0.151	0.802	–	0.0610	0.0027	0.0002	0.0089
	ACMDM-S-PS22+ProjFlow	✓	0.107	0.784	10.644	0.0629	**0.0000**	**0.0000**	**0.0000**
All Joints (Average)	OmniControl [61]	✗	0.126	0.792	10.276	0.0608	0.0617	0.0107	0.0404
	MotionLCM V2+CtrlNet [9]	✗	4.504	0.715	9.230	0.1119	0.2740	0.1315	0.2464
	MaskControl [45]	✗	0.095	0.795	10.159	**0.0545**	**0.0000**	**0.0000**	0.0065
	ACMDM-S-PS22+CtrlNet [39]	✗	**0.070**	**0.803**	**10.526**	0.0596	0.0117	0.0019	0.0197
	ACMDM-S-PS22+DNO [21]	✓	0.147	0.800	–	0.0600	0.0034	0.0003	0.0121
	ACMDM-S-PS22+ProjFlow	✓	0.097	0.779	10.651	0.0603	**0.0000**	**0.0000**	**0.0000**

	Methods	Zero-shot?	FID↓	R-Precision Top 1	R-Precision Top 2	R-Precision Top 3	Matching↓	Diversity→	–
Upper-Body Edit	MDM [56]	✓	1.918	0.359	0.556	0.654	4.793	9.210	–
	OmniControl [61]	✗	0.909	0.428	0.614	0.722	3.694	10.207	–
	MotionLCM V2+CtrlNet [9]	✗	3.922	0.404	0.592	0.692	5.610	9.309	–
	MaskControl [45]	✗	**0.066**	0.501	0.695	0.794	3.227	10.159	–
	ACMDM-S-PS22+CtrlNet [39]	✗	0.076	**0.532**	**0.719**	**0.820**	**3.098**	10.586	–
	ACMDM-S-PS22+ProjFlow	✓	0.087	0.501	0.690	0.787	3.319	**10.571**	–

based controllers such as OmniControl [61]. Even when the same base model is additionally trained with a ControlNet branch (ACMDM-S-PS22+CtrlNet), the constraints are still not fully satisfied, despite a slightly improved FID of 0.067. In contrast, ProjFlow achieves exact constraint satisfaction without any retraining.

Qualitative Analysis. Fig. 4 compares the generated motions from OmniControl [61], MaskControl [45], and ProjFlow. OmniControl [61] captures the overall S-shaped tendency of the target path but deviates significantly along the curve, especially near the bends. MaskControl [45] uses a ControlNet-style branch and additionally performs inference-time optimization, which further reduces this deviation. However, close inspection of the overlaid trajectories still reveals slight mismatches between the generated and target paths. By contrast, ProjFlow aligns the generated pelvis trajectory with the target markers essentially exactly across the entire S-shaped path while preserving natural full-body motion.

5.2.2. 2D-to-3D Reconstruction

Quantitative Performance. As shown in Table 2, ProjFlow achieves superior motion naturalness, attaining a lower FID than the state-of-the-art method Sketch2Anim [67] under both Average and Cross evaluation protocols. For constraint satisfaction, ProjFlow enforces the 2D constraints *exactly* to the numerical precision (MPJPE-2D $=$ $\mathbf{0.000}$) while Sketch2Anim [67] still exhibits residual reprojection errors.

Qualitative Analysis. Fig. 5 shows a qualitative example of the 2D-to-3D lifting task. The goal is to generate a 3D motion that follows the given 2D heart-shaped wrist trajectory and the given initial 2D keypose, while simultaneously "walking" as specified by the text prompt.

ProjFlow succeeds in following the 2D heart trajectory exactly at every frame while keeping the other joints engaged in a natural walking motion. The legs and torso continue to produce smooth, coordinated gait cycles as the left wrist draws the heart shape in the image plane. In contrast, Sketch2Anim [67] fails to preserve the heart shape, and the trajectory collapses into a distorted loop. The character also primarily remains in place, only moving the arm without translating forward, indicating that the intended instruction to walk is not realized.

5.2.3. Ablation Study

We analyze the contribution of ProjFlow's three key components on the motion inpainting task in Table 3. First, replacing our kinematics-aware metric with a standard Euclidean metric severely degrades motion realism, causing a significant degradation in FID. This confirms that propagating corrections coherently along the skeleton is critical. Second, removing the stochastic recomposition step ($\eta_t = 0$) and deterministically recomposing the state also drastically harms quality and diversity. This highlights the importance

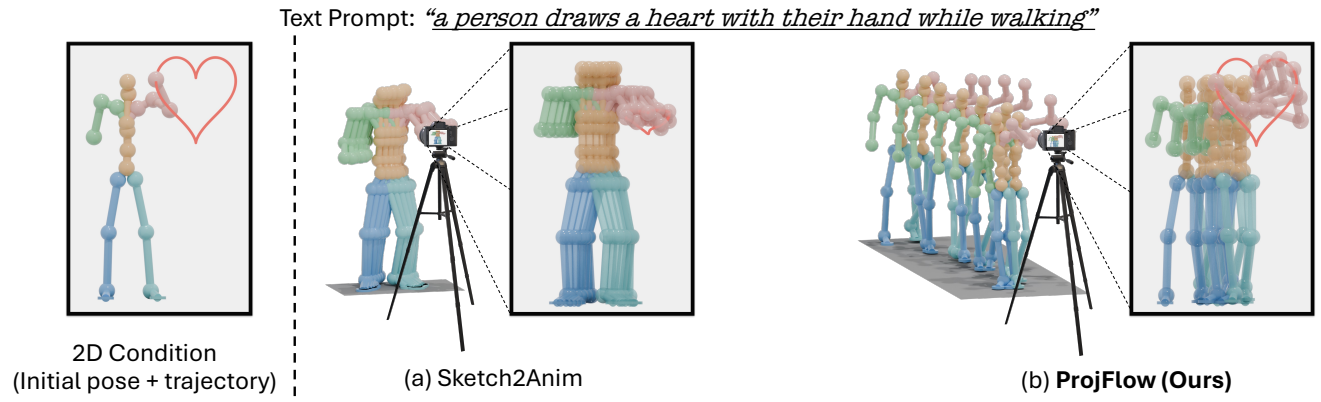


Figure 5. **2D-to-3D hand-trajectory lifting with text conditioning.** The input condition includes the text prompt "*a person draws a heart with their hand while walking*," an initial 2D keypose, and a left-wrist 2D trajectory shaped like a heart. Sketch2Anim [67] fails to reproduce the heart path precisely, the shape collapses, and the subject does not exhibit walking motion. In contrast, ProjFlow follows the heart-shaped wrist trajectory accurately while maintaining a natural walking motion throughout the sequence.

Table 2. **Quantitative analysis of ProjFlow and three baseline models proposed in Sketch2Anim [67] on the HumanML3D [16].** Evaluation metrics on motion realism, control accuracy, and text-motion match are presented. Following OmniControl [61], we report both the average error of all joints (Average) and their random combination (Cross). **bold** face / underline indicates the best/2nd results.

Condition	Method	Realism		Control Accuracy				Text-Motion Matching	
		FID ↓	Foot Skating ↓	MPJPE-2D ↓	MPJPE-3D ↓	Avg. Err.-2D ↓	Avg. Err.-3D ↓	Matching↓	R-precision (Top-3) ↑
Average	Motion Retrieval	0.690	**0.064**	0.057	0.076	0.290	0.410	4.060	0.640
	Lift-and-Control	0.979	0.089	0.054	0.071	0.261	0.340	3.297	0.752
	Direct 2D-to-Motion	2.553	0.112	0.040	0.055	0.193	0.275	3.723	0.687
	Sketch2Anim [67]	0.525	0.103	0.036	0.048	0.087	**0.134**	**3.077**	**0.802**
	ACMDM-S-PS22+ProjFlow	**0.349**	0.146	**0.000**	**0.042**	**0.000**	0.331	3.363	0.748
Cross	Motion Retrieval	**0.103**	**0.067**	0.055	0.073	0.307	0.423	3.405	0.724
	Lift-and-Control	0.738	0.101	0.051	0.067	0.209	0.283	3.135	0.778
	Direct 2D-to-Motion	2.310	0.123	0.040	0.056	0.189	0.266	3.606	0.709
	Sketch2Anim [67]	0.577	0.102	0.033	0.046	0.079	**0.132**	**3.042**	**0.796**
	ACMDM-S-PS22+ProjFlow	0.168	0.139	**0.000**	**0.037**	**0.000**	0.298	3.259	0.764

Table 3. Ablation studies of ProjFlow.

Variant	FID↓	R-Prec.	Div.→	Foot↓	Traj.↓	Loc.↓	Avg.↓
ProjFlow (Full)	0.097	0.779	10.651	0.0603	0.0000	0.0000	0.0000
Euclid. (R=I)	1.152	0.740	10.107	0.0595	0.0000	0.0000	0.0000
No noise (η_t=0)	3.429	0.707	9.307	0.0863	0.0000	0.0000	0.0000
Plain masking	0.880	0.748	10.187	0.0632	0.0000	0.0000	0.0000

of noise mixing for staying on the learned motion manifold. Third, for the inpainting task, reverting to a "Plain masking" approach without our pseudo-observation significantly worsens realism. These results validate that while all variants maintain exact constraint satisfaction, all three proposed components are essential for generating natural and realistic motion.

6. Limitations

While ProjFlow offers exact satisfaction of linear spatial constraints in a training-free manner, it is fundamentally limited to constraints that can be formulated as linear inverse problems. Our framework, in its current form, cannot natively handle more complex non-linear constraints. Examples of such constraints include inequalities such as keeping a joint above a certain plane. Extending the closed-form projection to these more expressive, non-linear scenarios is a challenging but important direction for future work.

7. Conclusion

In this paper, we presented **ProjFlow**, a zero-shot projection sampler for flow-matching models that achieves exact spatial motion control. Our method unifies diverse animation tasks, such as trajectory following and 2D-to-3D lifting, by formulating them as linear inverse problems. The sampler projects the clean motion estimate onto the linear constraint set at each ODE step. This projection employs a novel kinematics-aware metric that respects skeletal topology to maintain motion naturalness. ProjFlow successfully enforces hard constraints exactly without requiring any task-specific retraining or iterative optimization. Experiments on motion inpainting and 2D-to-3D reconstruction show that our framework matches the realism of training-based methods while guaranteeing exact constraint satisfaction. ProjFlow provides a practical route for interactive and precise motion authoring.

References

[1] Dhruv Agrawal, Jakob Buhmann, Dominik Borer, Robert W. Sumner, and Martin Guay. Skel-betweener: a neural motion rig for interactive motion authoring. *ACM Trans. Graph.*, 43 (6), 2024. 1

[2] Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations (ICLR)*, 2023. 3

[3] Lucas N. Alegre, Agon Serifi, Ruben Grandia, David Müller, Espen Knoop, and Moritz Bächer. Amor: Adaptive character control through multi-objective reinforcement learning. In *SIGGRAPH 2025 Conference Papers*. ACM, 2025. 2

[4] Manolo Canales Cuba, Vinícius do Carmo Melício, and João Paulo Gois. Flowmotion: Target-predictive conditional flow matching for jitter-reduced text-driven human motion generation. *arXiv preprint arXiv:2504.01338*, 2025. 2

[5] Junuk Cha, Jihyeon Kim, Jae Shin Yoon, and Seungryul Baek. Text2hoi: Text-guided 3d motion generation for hand-object interaction. In *CVPR*, 2024. 2

[6] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023. 3

[7] Setareh Cohan, Daniele Reda, Guy Tevet, Xue Bin Peng, and Michiel van de Panne. Flexible motion in-betweening with diffusion models. In *SIGGRAPH 2024 Conference Papers*. ACM, 2024. 2

[8] Rishabh Dabral, Muhammad Hamza Mughal, Vladislav Golyanik, and Christian Theobalt. Mofusion: A framework for denoising-diffusion-based motion synthesis. In *CVPR*, 2023. 2

[9] Wenxun Dai et al. Motionlcm: Real-time controllable motion generation via latent consistency models. *arXiv preprint arXiv:2404.19759*, 2024. 1, 2, 3, 7

[10] Christian Diller and Angela Dai. Cg-hoi: Contact-guided 3d human-object interaction generation. In *CVPR*, 2024. 2

[11] Markos Diomataris, Nikos Athanasiou, Omid Taheri, Xi Wang, Otmar Hilliges, and Michael J. Black. Wandr: Intention-guided human motion generation. In *CVPR*, 2024. 2

[12] Yuming Du, Robin Kips, Albert Pumarola, Sebastian Starke, Ali Thabet, and Artsiom Sanakoyeu. Avatars grow legs: Generating smooth human motion from sparse tracking inputs with diffusion models. In *CVPR*, 2023. 2

[13] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pages 12606–12633, Vienna, Austria, 2024. PMLR. 2

[14] Ke Fan, Junshu Tang, Weijian Cao, Ran Yi, Moran Li, Jingyu Gong, Jiangning Zhang, Yabiao Wang, Chengjie Wang, and Lizhuang Ma. Freemotion: A unified framework for number-free text-to-motion synthesis. In *ECCV*, 2024. 2

[15] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2021–2029, 2020. 6

[16] 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 (CVPR)*, pages 5152–5161, 2022. 2, 6, 7, 8

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pages 6840–6851. Curran Associates, Inc., 2020. 2

[18] Vincent Tao Hu, Wenzhe Yin, Pingchuan Ma, Yunlu Chen, Basura Fernando, Yuki M. Asano, Efstratios Gavves, Pascal Mettes, Björn Ommer, and Cees G. M. Snoek. Motion flow matching for human motion synthesis and editing. *arXiv preprint arXiv:2312.08895*, 2023. 2

[19] Siyuan Huang, Zan Wang, Puhao Li, Baoxiong Jia, Tengyu Liu, Yixin Zhu, Wei Liang, and Song-Chun Zhu. Diffusion-based generation, optimization, and planning in 3d scenes. In *CVPR*, 2023. 2

[20] Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. Guided motion diffusion for controllable human motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 21510–21522, 2023. 2, 3, 7

[21] Korrawe Karunratanakul, Konpat Preechakul, Emre Aksan, Thabo Beeler, Supasorn Suwajanakorn, and Siyu Tang. Optimizing diffusion noise can serve as universal motion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1334–1345, 2024. 1, 3, 6, 7

[22] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *Advances in Neural Information Processing Systems*, 2022. 3

[23] Jeongsol Kim, Bryan Sangwoo Kim, and Jong Chul Ye. FlowDPS: Flow-driven posterior sampling for inverse problems, 2025. 4

[24] Nilesh Kulkarni, Davis Rempe, Kyle Genova, Abhijit Kundu, Justin Johnson, David Fouhey, and Leonidas Guibas. Nifty: Neural object interaction fields for guided human motion synthesis. In *CVPR*, 2024. 2

[25] Hsin-Ying Lee, Xiaodong Yang, Ming-Yu Liu, Ting-Chun Wang, Yu-Ding Lu, Ming-Hsuan Yang, and Jan Kautz. Dancing to music. In *NeurIPS*, 2019. 2

[26] Buyu Li, Yongchi Zhao, Zhelun Shi, and Lu Sheng. Danceformer: Music conditioned 3d dance generation with parametric motion transformer. In *AAAI*, 2021. 2

[27] Jiaman Li, Alexander Clegg, Roozbeh Mottaghi, Jiajun Wu, Xavier Puig, and C. Karen Liu. Controllable human-object interaction synthesis. *arXiv:2312.03913*, 2023. 2

[28] Ruilong Li, Sha Yang, David A. Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *ICCV*, 2021. 2

[29] Siyao Li, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, and Ziwei Liu. Bailando:

3d dance generation by actor-critic gpt with choreographic memory. In *CVPR*, 2022.

[30] Siyao Li, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, and Ziwei Liu. Bailando++: 3d dance gpt with choreographic memory. *IEEE TPAMI*, 2023. 2

[31] Han Liang, Jiacheng Bao, Ruichi Zhang, Sihan Ren, Yuecheng Xu, Sibei Yang, Xin Chen, Jingyi Yu, and Lan Xu. Omg: Towards open-vocabulary motion generation via mixture of controllers. In *CVPR*, 2024. 3

[32] Han Liang, Wenqian Zhang, Wenxuan Li, Jingyi Yu, and Lan Xu. Intergen: Diffusion-based multi-human motion generation under complex interactions. *IJCV*, pages 1–21, 2024. 2

[33] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. 2, 3

[34] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code, 2024. 2, 3

[35] Hanchao Liu, Xiaohang Zhan, Shaoli Huang, Tai-Jiang Mu, and Ying Shan. Programmable motion generation for open-set motion control tasks. In *CVPR*, 2024. 2

[36] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023. 2, 3

[37] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451, 2019. 6

[38] Ségolène Tiffany Martin, Anne Gagneux, Paul Hagemann, and Gabriele Steidl. Pnp-flow: Plug-and-play image restoration with flow matching. In *The Thirteenth International Conference on Learning Representations*, 2025. 3

[39] Zichong Meng, Zeyu Han, Xiaogang Peng, Yiming Xie, and Huaizu Jiang. Absolute coordinates make motion generation easy. *arXiv preprint arXiv:2505.19377*, 2025. 1, 2, 6, 7

[40] Zichong Meng, Yiming Xie, Xiaogang Peng, Zeyu Han, and Huaizu Jiang. Rethinking diffusion for text-driven human motion generation: Redundant representations, evaluation, and masked autoregression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27859–27871, 2025. 2, 6

[41] Sakuya Ota, Qing Yu, Kent Fujiwara, Satoshi Ikehata, and Ikuro Sato. Pino: Person-interaction noise optimization for long-duration and customizable motion generation of arbitrary-sized groups. In *ICCV*, 2025. 2

[42] Maitreya Patel, Song Wen, Dimitris N. Metaxas, and Yezhou Yang. Flowchef: Steering of rectified flow models for controlled generations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15308–15318, 2025. 3

[43] Mathis Petrovich, Or Litany, Umar Iqbal, Michael J. Black, Gul Varol, Xue Bin Peng, and Davis Rempe. Multi-track timeline control for text-driven 3d human motion generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1911–1921, 2024. 2

[44] Huaijin Pi, Zhi Cen, Zhiyang Dou, and Taku Komura. Coda: Coordinated diffusion noise optimization for whole-body manipulation of articulated objects. *arXiv preprint arXiv:2505.21437*, 2025. 1

[45] Ekkasit Pinyoanuntapong, Muhammad Usama Saleem, Korrawe Karunratanakul, Pu Wang, Hongfei Xue, Chen Chen, Chuan Guo, Junli Cao, Jian Ren, and Sergey Tulyakov. Controlmm: Controllable masked motion generation. *arXiv preprint arXiv:2410.10780*, 2024. 1, 3, 6, 7

[46] Davis Rempe, Zhengyi Luo, Xue Bin Peng, Ye Yuan, Kris Kitani, Karsten Kreis, Sanja Fidler, and Or Litany. Trace and pace: Controllable pedestrian animation via guided trajectory diffusion. In *CVPR*, 2023. 3

[47] Roni Ron, Guy Tevet, Harel Sawdayee, and Amit H. Bermano. Hoidini: Human-object interaction through diffusion noise optimization. *arXiv preprint arXiv:2506.15625*, 2025. 1

[48] Litu Rout, Negin Raoof, Giannis Daras, Constantine Caramanis, Alex Dimakis, and Sanjay Shakkottai. Solving linear inverse problems provably via posterior sampling with latent diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 3

[49] Agon Serifi, Ruben Grandia, Espen Knoop, Markus Gross, and Moritz Bächer. Robot motion diffusion model: Motion generation for robotic characters. In *SIGGRAPH Asia 2024 Conference Papers*. ACM, 2024. 2

[50] Yoni Shafir, Guy Tevet, Roy Kapon, and Amit H. Bermano. Human motion diffusion as a generative prior. In *ICLR*, 2024. 2, 7

[51] Bowen Song, Soo Min Kwon, Zecheng Zhang, Xinyu Hu, Qing Qu, and Liyue Shen. Solving inverse problems with latent diffusion models via hard data consistency. In *The Twelfth International Conference on Learning Representations*, 2024. 3

[52] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023. 3

[53] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021. 2

[54] Justin Studer, Dhruv Agrawal, Dominik Borer, Seyedmorteza Sadat, Robert W. Sumner, Martin Guay, and Jakob Buhmann. Factorized motion diffusion for precise and character-agnostic motion inbetweening. In *Proceedings of the 17th ACM SIGGRAPH Conference on Motion, Interaction, and Games*, New York, NY, USA, 2024. Association for Computing Machinery. 1

[55] Mikihiro Tanaka and Kent Fujiwara. Role-aware interaction generation from textual description. In *ICCV*, 2023. 2

[56] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-Or, and Amit Haim Bermano. Human motion diffusion model. In *International Conference on Learning Representations (ICLR)*, 2023. 2, 7

[57] Jo-Han Tseng, Rodrigo Castellon, and C. Karen Liu. Edge: Editable dance generation from music. In *CVPR*, 2022. 2

[58] Weilin Wan, Zehao Dou, Taku Komura, Wenping Wang, Dinesh Jayaraman, and Lingjie Liu. Tlcontrol: Trajectory and language control for human motion synthesis. In *ECCV*, 2024. 3

[59] Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. In *The Eleventh International Conference on Learning Representations*, 2023. 3

[60] Zan Wang, Yixin Chen, Baoxiong Jia, Puhao Li, Jinlu Zhang, Jingze Zhang, Tengyu Liu, Yixin Zhu, Wei Liang, and Siyuan Huang. Move as you say, interact as you can: Language-guided human motion generation with scene affordance. In *CVPR*, 2024. 2

[61] Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. Omnicontrol: Control any joint at any time for human motion generation. In *International Conference on Learning Representations (ICLR)*, 2024. 1, 2, 3, 6, 7, 8

[62] Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. Improving diffusion inverse problem solving with decoupled noise annealing. *arXiv preprint arXiv:2407.01521*, 2024. 3

[63] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3813–3824, 2023. 3

[64] Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Remodiffuse: Retrieval-augmented motion diffusion model. In *ICCV*, pages 364–373, 2023. 2

[65] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Text-driven human motion generation with diffusion model. *IEEE TPAMI*, 2024. 2

[66] Lei Zhong, Yiming Xie, Varun Jampani, Deqing Sun, and Huaizu Jiang. Smoodi: Stylized motion diffusion model. *arXiv:2407.12783*, 2024. 2

[67] Lei Zhong, Chuan Guo, Yiming Xie, Jiawei Wang, and Changjian Li. Sketch2anim: Towards transferring sketch storyboards into 3d animation. *ACM Transactions on Graphics*, 44(4):1–15, 2025. 2, 6, 7, 8