

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

## Supplementary Material

This supplementary material is organized as follows:

- Section A: Analytical view of ProjFlow.
- Section B: Additional method details.
- Section C: Implementation details.
- Section D: Additional quantitative results.
- Section E: Additional qualitative results.

### A. Analytical View of ProjFlow

Using the notation in Table 4, we provide an analytical interpretation of the ProjFlow update, including its relation to DDNM and a MAP view. In what follows, “PSD” and “PD” denote positive semidefinite and positive definite matrices, respectively.

Table 4. Notation used in the supplementary derivations.

Symbol	Type / shape	Note
$A$	$\mathbb{R}^{m \times d}$	linear operator
$\mathbf{y}$	$\mathbb{R}^m$	measurements
$\mathbf{x}_1$	$\mathbb{R}^d$	clean motion endpoint
$\hat{\mathbf{x}}_1$	$\mathbb{R}^d$	estimate of $\mathbf{x}_1$
$\Sigma$	$\mathbb{R}^{m \times m}$	PSD covariance
$R$	$\mathbb{R}^{d \times d}$	PD metric / precision

#### A.1. Recovery of DDNM under Euclidean Metric and Noiseless Observation

DDNM [60] solves the linear inverse problem  $\mathbf{y} = A\mathbf{x}$  by decomposing  $\mathbb{R}^d$  into the range and null space of  $A$ . Given a clean-endpoint estimate  $\hat{\mathbf{x}}_1$ , it keeps the range-space component consistent with the measurements and fills the null space with  $\hat{\mathbf{x}}_1$ :

$$\hat{\mathbf{x}}_1^* = A^\dagger \mathbf{y} + (I - A^\dagger A) \hat{\mathbf{x}}_1, \quad (21)$$

where  $A^\dagger$  is the Moore–Penrose pseudoinverse of  $A$ .

ProjFlow, in contrast, updates the clean-endpoint estimate via

$$\hat{\mathbf{x}}_1^* = \hat{\mathbf{x}}_1 + R^{-1} A^\top (A R^{-1} A^\top + \Sigma)^{-1} (\mathbf{y} - A \hat{\mathbf{x}}_1). \quad (22)$$

Specializing to the Euclidean metric  $R = I$  and the noise-free limit  $\Sigma \rightarrow 0$ , and assuming that  $A$  has full row rank so that  $AA^\top$  is invertible, we obtain

$$\hat{\mathbf{x}}_1^* = \hat{\mathbf{x}}_1 + A^\top (AA^\top)^{-1} (\mathbf{y} - A \hat{\mathbf{x}}_1) \quad (23)$$

$$= \hat{\mathbf{x}}_1 + A^\dagger \mathbf{y} - A^\dagger A \hat{\mathbf{x}}_1 \quad (24)$$

$$= A^\dagger \mathbf{y} + (I - A^\dagger A) \hat{\mathbf{x}}_1, \quad (25)$$

which coincides exactly with the DDNM update above. Thus, DDNM is recovered as a special case of ProjFlow in the Euclidean, noiseless setting.

#### A.2. ProjFlow as MAP Estimation

ProjFlow’s projection step can also be interpreted as computing a maximum-a-posteriori (MAP) estimate in a linear–Gaussian model. We treat the clean-endpoint estimate  $\hat{\mathbf{x}}_1$  from Tweedie’s formula as the mean of a Gaussian prior

$$p(\mathbf{x}_1) = \mathcal{N}(\mathbf{x}_1 | \hat{\mathbf{x}}_1, R^{-1}), \quad (26)$$

where  $R \succ 0$  is the precision matrix and  $R^{-1}$  is the corresponding covariance.

For the Euclidean metric  $R = I$ , this prior is an isotropic Gaussian centered at  $\hat{\mathbf{x}}_1$ , penalizing all directions equally. With the kinematics-aware metric  $R$ , the structure is instead governed by the skeletal Laplacian  $L_{\text{kin}}$ : directions that create large differences between adjacent joints (skeletonally incoherent motion) have small variance, while coordinated joint motions have larger variance. Geometrically, this yields a highly anisotropic ellipsoidal prior that favors kinematically coherent corrections.

The linear observation model is

$$\mathbf{y} = A\mathbf{x}_1 + \epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \Sigma) \quad (27)$$

$$\iff p(\mathbf{y} | \mathbf{x}_1) = \mathcal{N}(\mathbf{y} | A\mathbf{x}_1, \Sigma). \quad (28)$$

Combining this likelihood with the prior yields a Gaussian posterior

$$p(\mathbf{x}_1 | \mathbf{y}) \propto \exp\left(-\frac{1}{2}\|\mathbf{x}_1 - \hat{\mathbf{x}}_1\|_R^2 - \frac{1}{2}\|\mathbf{y} - A\mathbf{x}_1\|_{\Sigma^{-1}}^2\right). \quad (29)$$

The MAP estimate  $\mathbf{x}_1^{\text{MAP}}$  maximizes this posterior, or equivalently minimizes the negative log-posterior:

$$\mathbf{x}_1^{\text{MAP}} = \arg \min_{\mathbf{x}_1} \left( \|\mathbf{x}_1 - \hat{\mathbf{x}}_1\|_R^2 + \|\mathbf{y} - A\mathbf{x}_1\|_{\Sigma^{-1}}^2 \right). \quad (30)$$

Taking the gradient with respect to  $\mathbf{x}_1$  and setting it to zero gives the normal equations

$$(R + A^\top \Sigma^{-1} A) \mathbf{x}_1 = R \hat{\mathbf{x}}_1 + A^\top \Sigma^{-1} \mathbf{y}, \quad (31)$$

so that

$$\mathbf{x}_1^{\text{MAP}} = (R + A^\top \Sigma^{-1} A)^{-1} (R \hat{\mathbf{x}}_1 + A^\top \Sigma^{-1} \mathbf{y}) \quad (32)$$

$$= \hat{\mathbf{x}}_1 + (R + A^\top \Sigma^{-1} A)^{-1} A^\top \Sigma^{-1} (\mathbf{y} - A \hat{\mathbf{x}}_1). \quad (33)$$

The second line makes explicit that the MAP solution is obtained by adding a correction to  $\hat{\mathbf{x}}_1$ . Using standard linear–Gaussian identities, this correction term is equivalent to the

886 ProjFlow update  
887  $\hat{\mathbf{x}}_1^* = \hat{\mathbf{x}}_1 + R^{-1}A^\top (AR^{-1}A^\top + \Sigma)^{-1}(\mathbf{y} - A\hat{\mathbf{x}}_1), \quad (34)$

888 showing that ProjFlow’s projection step is exactly the MAP  
889 estimate of this linear–Gaussian model.

## 890 B. Additional Method Details

### 891 B.1. Formulating Teaser Applications as Linear In- 892 verse Problems

893 We briefly show how the additional teaser applications in  
894 Fig. 1 fit into the unified linear model  $\mathbf{y} = A\mathbf{x} + \epsilon$ . Tra-  
895 jectory control and 2D-to-3D lifting are already described  
896 in the main paper. Here, we detail the relative position con-  
897 straint and looped motion.

#### 898 B.1.1. Relative Position Constraint

899 We consider the case where the relative 3D position be-  
900 tween two joints remains fixed, e.g., both wrists holding a  
901 rigid object. Let  $\mathbf{x}_{n,j_a}, \mathbf{x}_{n,j_b} \in \mathbb{R}^3$  denote the positions of  
902 joints  $j_a$  and  $j_b$  at frame  $n$ . To keep their 3D offset fixed,  
903 we enforce for each frame

$$904 \mathbf{x}_{n,j_a} - \mathbf{x}_{n,j_b} = \mathbf{d}, \quad (35)$$

905 where  $\mathbf{d} = (d_x, d_y, d_z)^\top$  is the desired 3D offset vector.  
906 This is linear in the full motion vector  $\mathbf{x}$ . Stacking the con-  
907 straints over all  $N$  frames yields a standard linear inverse  
908 problem

$$909 \mathbf{y}_{\text{rel}} = A_{\text{rel}}\mathbf{x}, \quad (36)$$

910 where  $\mathbf{y}_{\text{rel}}$  is  $\mathbf{d}$  repeated  $N$  times, so  $\mathbf{y}_{\text{rel}} \in \mathbb{R}^{3N}$ . The  
911 operator  $A_{\text{rel}} \in \mathbb{R}^{3N \times d}$  is a sparse matrix that, for each  
912 frame, subtracts the coordinates of joint  $j_b$  from those of  
913 joint  $j_a$ .

#### 914 B.1.2. Looped Motion

915 To make a sequence loop seamlessly, we match the start and  
916 end poses. Let  $\mathbf{x}_0$  and  $\mathbf{x}_{N-1}$  be the first and last frames of  
917 the motion, respectively. We impose the per-joint constraint

$$918 \mathbf{x}_0 - \mathbf{x}_{N-1} = \mathbf{0}, \quad (37)$$

919 which is again linear in  $\mathbf{x}$ . Stacking these equations over all  
920 joints and spatial coordinates gives

$$921 \mathbf{0} = A_{\text{loop}}\mathbf{x}, \quad (38)$$

922 where  $A_{\text{loop}} \in \mathbb{R}^{3J \times d}$  computes the difference between the  
923 first and last frames. In our framework, this loop-closure  
924 operator can simply be concatenated with other linear con-  
925 straints by stacking its rows into the global observation ma-  
926 trix  $A$ .

## 511 B.2. Detailed Formulation of Motion Inpainting 927

### 512 B.2.1. Pseudo-observations: linear interpolation and ex- 513 trapolation 929

514 We generate pseudo-observations by per-joint linear inter-  
515 polation. For each joint, we scan all unobserved frames  
516 and, for a given unobserved frame, locate the nearest ob-  
517 served frame before it and the nearest observed frame after  
518 it. If both exist, the frame lies between two known points,  
519 and we define the pseudo-observation by linear interpola-  
520 tion between these two observations. 935

521 If the frame lies outside the observed range for that joint  
522 (before the first observation or after the last), interpolation  
523 is impossible. In this case, we perform extrapolation by  
524 copying the value of the single nearest observed frame. If a  
525 joint has no observations at all in the sequence, we leave it  
526 without pseudo-observations. 942

### 527 B.2.2. Designing the adaptive variance 943

528 Our inpainting strategy augments sparse hard keyframe  
529 constraints with “soft” pseudo-observations from interpo-  
530 lation. The key challenge is to modulate the influence of  
531 these soft guides: they should be trusted less (i) at frames  
532 with high motion curvature, where interpolation is unreli-  
533 able, and (ii) late in sampling, when the model’s own pre-  
534 diction  $\hat{\mathbf{x}}_1$  is more reliable. We encode this behavior in a  
535 time-varying observation covariance  $\Sigma^{(t)}$ . Directly hand-  
536 designing variances  $\sigma_i^2(t)$  is unintuitive, so we instead de-  
537 sign a normalized trust score  $\pi_i \in [0, 1]$  and then derive the  
538 corresponding  $\sigma_i^2(t)$ . 954

539 To see the relation between  $\pi_i$  and  $\sigma_i^2(t)$ , we first con-  
540 sider a simple Euclidean case. For motion inpainting, the  
541 observation operator is a diagonal mask matrix  $A = M^{(t)}$ .  
542 Assuming the Euclidean metric  $R = I$ , the ProjFlow update  
543 becomes 959

$$544 \hat{\mathbf{x}}_1^* = \hat{\mathbf{x}}_1 + M^{(t)}(M^{(t)} + \Sigma^{(t)})^{-1}(\mathbf{y}^{(t)} - A\hat{\mathbf{x}}_1) \quad (39) \quad 960$$

$$545 = \left(I - M^{(t)}(M^{(t)} + \Sigma^{(t)})^{-1}M^{(t)}\right)\hat{\mathbf{x}}_1 \quad 961$$

$$546 + M^{(t)}(M^{(t)} + \Sigma^{(t)})^{-1}\mathbf{y}^{(t)}. \quad (40) \quad 962$$

547 In the inpainting setting, both  $M^{(t)}$  and  $\Sigma^{(t)}$  are diagonal,  
548 so this matrix equation decomposes into independent scalar  
549 updates. For an observed coordinate  $i$  (i.e.,  $M_{ii}^{(t)} = 1$ ) with  
550  $\Sigma_{ii}^{(t)} = \sigma_i^2(t)$ , we obtain 966

$$551 \hat{x}_{1,i} = \left(1 - \frac{1}{1 + \sigma_i^2(t)}\right)\hat{x}_{1,i} + \frac{1}{1 + \sigma_i^2(t)}y_i \quad (41) \quad 967$$

552 Thus, each updated coordinate is a weighted average of the  
553 model prediction  $\hat{x}_{1,i}$  and the observation  $y_i$ . If we define  
554 the weight on the observation as 970

$$555 \pi_{i,\text{Euclid}} \equiv \frac{1}{1 + \sigma_i^2(t)}, \quad (42) \quad 971$$

the update takes the intuitive form

$$\hat{x}_{1,i}^* = (1 - \pi_{i,\text{Euclid}}) \hat{x}_{1,i} + \pi_{i,\text{Euclid}} y_i. \quad (43)$$

This shows that, in the Euclidean case, the “weight on data” for an active coordinate is exactly  $\pi_{i,\text{Euclid}} = 1/(1 + \sigma_i^2(t))$ .

We now extend this idea to the kinematics-aware metric. The ProjFlow update becomes

$$\begin{aligned} \hat{x}_1^* &= \hat{x}_1 + R^{-1} M^{(t)\top} \left( M^{(t)} R^{-1} M^{(t)\top} + \Sigma^{(t)} \right)^{-1} \left( \mathbf{y}^{(t)} - M^{(t)} \hat{x}_1 \right) \\ &= \left( I - R^{-1} M^{(t)\top} \left( M^{(t)} R^{-1} M^{(t)\top} + \Sigma^{(t)} \right)^{-1} M^{(t)} \right) \hat{x}_1 \\ &\quad + R^{-1} M^{(t)\top} \left( M^{(t)} R^{-1} M^{(t)\top} + \Sigma^{(t)} \right)^{-1} \mathbf{y}^{(t)}. \end{aligned} \quad (44)$$

Here,  $R^{-1}$  is dense along joint dimensions, so corrections propagate across joints, while we still choose  $\Sigma^{(t)}$  to be diagonal, with each coordinate (frame–joint–axis) having its own variance. We therefore design a dimensionless trust score  $\pi_i \in [0, 1]$  for each active row  $i$ , and convert it into a variance that is consistent with the metric  $R$ .

Let  $r_i := [\text{diag}(R^{-1})]_i > 0$ . If only row  $i$  were active (i.e.,  $M^{(t)} = e_i^\top$ ), the measurement-space gain of the ProjFlow update is

$$\pi_i = \frac{r_i}{r_i + \sigma_i^2(t)}. \quad (46)$$

Solving for  $\sigma_i^2(t)$  yields

$$\Sigma_{ii}^{(t)} = \sigma_i^2(t) = r_i \left( \frac{1}{\pi_i} - 1 \right). \quad (47)$$

Note that when  $R = I$ , we have  $r_i = 1$ , and equation 47 reduces to  $\pi_i = 1/(1 + \sigma_i^2(t))$ , matching the Euclidean case.

### B.2.3. Computing the variance from the trust score for multiple joints

To obtain the per-element trust scores  $\pi_i$ , we first compute a frame-level base trust

$$\tilde{\pi}_n^{(t)} = \tau(t) \frac{c_0}{1 + \lambda_s (s_n(\hat{x}_1)/s_{\text{med}})^p}, \quad (48)$$

where  $n$  indexes frames,  $s_n(\hat{x}_1)$  is the curvature at frame  $n$ ,  $s_{\text{med}}$  is the median curvature over the sequence, and  $c_0, \lambda_s, p$  are hyperparameters. This  $\tilde{\pi}_n^{(t)}$  is the total “trust budget” for all active pseudo-observations in frame  $n$ . If only one joint has an active pseudo-observation at that frame, we simply set  $\pi_i = \tilde{\pi}_n^{(t)}$ .

If multiple joints are active in frame  $n$ , we distribute the frame-level budget across them according to their influence in the kinematics-aware metric  $R$ . Intuitively, we want to assign less trust to high-influence joints (e.g., pelvis) and more trust to low-influence joints (e.g., wrists). Let  $\mathcal{H}_n$  be the set of joints  $j$  with an active pseudo-observation in frame  $n$ , and  $m_n = |\mathcal{H}_n|$ . Recall that

$$R = w_{\text{kin}}(I_3 \otimes I_N \otimes L_{\text{kin}}) + \lambda I_d, \quad (49)$$

and define the joint-only component

$$R_J = w_{\text{kin}} L_{\text{kin}} + \lambda I_J. \quad (50)$$

From  $R_J$ , we define a per-joint weight as

$$q_j := \frac{1}{\|\mathbf{c}_j\|_2}, \quad (51)$$

where  $\mathbf{c}_j$  denotes the  $j$ -th column of  $R_J^{-1}$ . In other words,  $q_j$  is the reciprocal of the Euclidean norm of the  $j$ -th column of  $R_J^{-1}$ . Joints with large global influence yield columns with large norms and therefore smaller  $q_j$ , whereas low-influence joints yield smaller column norms and thus larger  $q_j$ .

We then distribute the frame budget proportionally to these weights. For an element  $i$  corresponding to joint  $j \in \mathcal{H}_n$ , we set

$$\pi_i = \text{clip} \left( \tilde{\pi}_n^{(t)} \frac{q_j}{\sum_{k \in \mathcal{H}_n} q_k}, \pi_{\min}, \pi_{\max} \right). \quad (52)$$

Ignoring clipping, this construction preserves the frame-level budget,  $\sum_{i \in \mathcal{H}_n} \pi_i = \tilde{\pi}_n^{(t)}$ , while assigning lower trust to high-influence joints and higher trust to low-influence ones. Finally, these  $\pi_i$  are converted to variances  $\sigma_i^2(t)$  via equation 47, yielding the diagonal entries of  $\Sigma^{(t)}$  for the active pseudo-observations.

## C. Implementation Details

### C.1. Application I: Motion Inpainting via Masked Pseudo-observations

The hyperparameters used for motion inpainting are summarized in Table 5. We use the same values for all inpainting experiments, including the main comparison in Table 1 and the ablation study in Table 3. We set the number of ODE sampling steps to  $T = 100$ , which corresponds to 100 function evaluations. The kinematics-aware metric  $R$  is parameterized with  $w_{\text{kin}} = 10.0$  and  $\lambda = 1.0$ . The dynamic masking radius shrinks linearly from  $l_{\text{max}} = 10$  to  $l_{\text{min}} = 3$  frames over time. For recomposition, we adopt the stochastic step from FlowDPS [23] with the noise-mixing schedule  $\eta_t = 1 - \sigma_{t+\Delta t}$ .

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

For the 2D-to-3D motion lifting application, we reuse the ODE sampler hyperparameters from the inpainting task. We again set  $T = 100$  sampling steps and use the FlowDPS [23] noise-mixing schedule  $\eta_t = 1 - \sigma_{t+\Delta t}$ . The kinematics-aware metric  $R$  also uses the same values  $w_{\text{kin}} = 10.0$  and  $\lambda = 1.0$  as in the inpainting experiments, without additional tuning for this task.

Block	Name	Symbol	Value
Kinematics-aware metric	joint coupling weight	$w_{\text{kin}}$	10.0
	ridge	$\lambda$	1.0
Dynamic Masking	min radius (frames)	$l_{\text{min}}$	3
	max radius (frames)	$l_{\text{max}}$	10
Adaptive Variance	time base	$\tau_{\text{min}}$	0.1
	strength	$c_0$	3.0
	curvature gain	$\lambda_s$	1.0
	curvature power	$p$	2.0
	trust clipping	$[\pi_{\text{min}}, \pi_{\text{max}}]$	[0.02, 1.0]
ODE sampling	NFE	$T$	100
	noise mixing	$\eta_t$	$\eta_t = 1 - \sigma_{t+\Delta t}$

Table 5. Hyperparameters for motion inpainting.

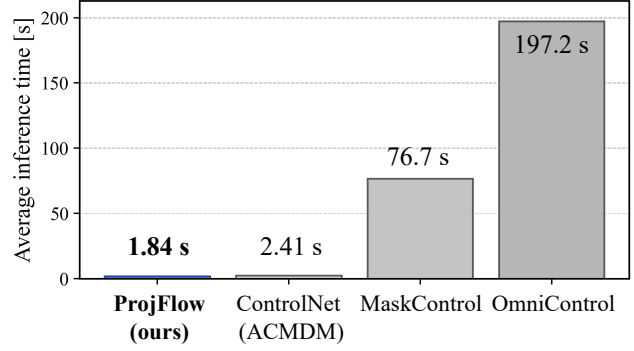


Figure 6. Average inference time per 196-frame sample .

## D. Additional Quantitative Results

### D.1. Inference Speed Comparison

We compare the inference speed of ProjFlow against training-based controllers that use the same backbone. Figure 6 reports the average wall-clock time required to generate one 196-frame motion sample on a single A100 GPU. The x-axis labels in the figure use abbreviated names: ProjFlow (ours) and ControlNet (ACMDM) correspond to ACMDM-S-PS22+ProjFlow and ACMDM-S-PS22+ControlNet, respectively. We use the original settings from each paper whenever they are specified. For ControlNet [39], whose sampling schedule is not detailed, we match our ProjFlow configuration for fairness: both ProjFlow and ControlNet use 100 Euler steps. OmniControl [63] is evaluated with 1,000 sampling steps. MaskControl [45] uses 10 sampling steps, with 100 logits-optimization steps at each unmasking step and 600 optimization steps at the final unmasking step as described in the original paper.

Under these settings, ProjFlow achieves an average inference time of **1.84 s** per sample and is the fastest among all compared methods. Notably, even though ProjFlow and ControlNet (ACMDM) share the same 100-step sampling schedule, ProjFlow runs faster because it keeps the original backbone unchanged, whereas ControlNet attaches an additional conditioning branch that increases model size and inference cost.

### D.2. Detailed Results of Ablation Study

Figure 7 summarizes how each ProjFlow component affects robustness to control intensity on the motion inpainting task. The model Full is compared with three ablations: Euclid, which uses a standard Euclidean metric instead of the kinematics-aware metric; Without Noise, which removes the stochastic noise-mixing step; and Plain Masking, which removes pseudo-observations and uses only hard keyframes. When observations are sparse, both the Euclidean metric and the deterministic recomposition

(Without Noise) noticeably degrade realism, and the Plain Masking variant performs worst, confirming the importance of our pseudo-observations. Table 6 reports the full per-joint numbers: our Full model consistently attains the best FID and R-Precision across all controlled joints, while keeping trajectory, location, and average control errors at zero.

### D.3. Detailed Results of Motion Inpainting

In the main paper Table 1, we presented a summarized version of the controllable motion generation results. Table 7 provides the complete per-joint evaluation, following the OmniControl [63] protocol. Across all controlled joints, ProjFlow achieves zero trajectory, location, and average errors, while its FID, R-Precision, and diversity scores remain in the same band as the strongest training-based controllers. This shows that enforcing exact spatial constraints with ProjFlow does not come at the expense of motion realism.

### D.4. Evaluation on Legacy Metrics

Meng et al. [40] recently highlighted several shortcomings in the conventional HumanML3D evaluation protocol and proposed revised metrics, which we adopt for the main results in the paper. However, many prior works [9, 20, 45, 46, 51, 57, 63] still report performance using the legacy protocol, making direct comparison otherwise impossible. To broaden the set of comparable baselines, we therefore also evaluate ProjFlow and existing methods under the original evaluation setup. The results are summarized in Table 8. Under this legacy protocol, ProjFlow remains competitive with strong training-based controllers while retaining its zero-shot nature and exact constraint satisfaction.

## E. Additional qualitative results.

The supplementary material includes a browsable demo page that collects our qualitative videos (open [index.html](#) in a web browser). This page organizes examples by task: the four control scenarios from Fig.1,

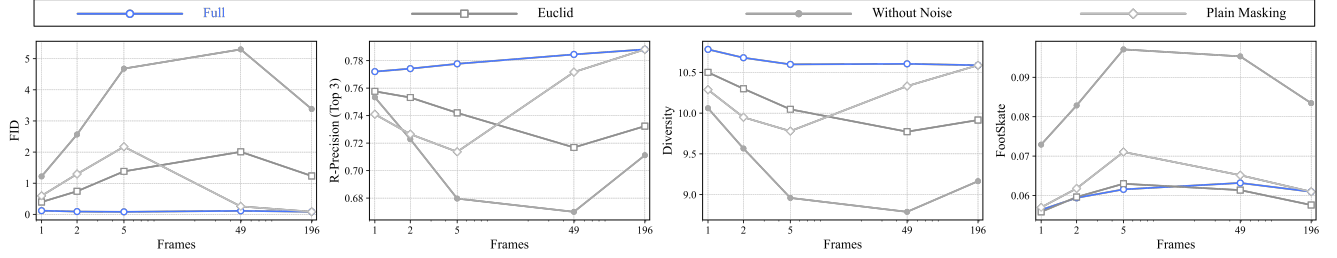


Figure 7. **Ablation of ProjFlow components vs. control intensity on motion inpainting.** We compare the full model (Full) against variants that (i) replace the kinematics-aware metric with a Euclidean metric (Euclid), (ii) remove the stochastic noise-mixing step (Without Noise), or (iii) disable pseudo-observations and rely only on hard keyframes (Plain Masking).

Table 6. **Ablation of ACMDM-S-PS22+ProjFlow on HumanML3D.** Methods are evaluated on all joints and reported per controlled joint. **bold face** / underline indicates the best/2<sup>nd</sup> results if applied.

Controlling Joint	Methods	FID↓	R-Precision Top 3	Diversity→	Foot Skating Ratio↓	Traj. err.↓	Loc. err.↓	Avg. err.↓
	GT	0.000	0.795	10.455	-	0.0000	0.0000	0.0000
Pelvis	ACMDM-S-PS22+ProjFlow (Full)	<b>0.107</b>	<b>0.784</b>	<b>10.645</b>	<u>0.0630</u>	0.0000	0.0000	0.0000
	ACMDM-S-PS22+ProjFlow (Euclid)	4.360	0.686	8.953	<b>0.0550</b>	0.0000	0.0000	0.0000
	ACMDM-S-PS22+ProjFlow (w/o noise)	2.439	<u>0.734</u>	9.666	0.0960	0.0000	0.0000	0.0000
	ACMDM-S-PS22+ProjFlow (Plain Masking)	<u>2.091</u>	0.726	<u>9.838</u>	0.0658	0.0000	0.0000	0.0000
Left foot	ACMDM-S-PS22+ProjFlow (Full)	<b>0.095</b>	<b>0.771</b>	<b>10.644</b>	<b>0.0609</b>	0.0000	0.0000	0.0000
	ACMDM-S-PS22+ProjFlow (Euclid)	<u>0.476</u>	0.743	<u>10.399</u>	<u>0.0643</u>	0.0000	0.0000	0.0000
	ACMDM-S-PS22+ProjFlow (w/o noise)	4.450	0.680	9.209	0.0969	0.0000	0.0000	0.0000
	ACMDM-S-PS22+ProjFlow (Plain Masking)	0.576	<u>0.746</u>	10.309	0.0681	0.0000	0.0000	0.0000
Right foot	ACMDM-S-PS22+ProjFlow (Full)	<b>0.096</b>	<b>0.770</b>	<b>10.651</b>	<b>0.0613</b>	0.0000	0.0000	0.0000
	ACMDM-S-PS22+ProjFlow (Euclid)	<u>0.486</u>	0.745	<u>10.359</u>	<u>0.0655</u>	0.0000	0.0000	0.0000
	ACMDM-S-PS22+ProjFlow (w/o noise)	4.805	0.673	9.129	0.0944	0.0000	0.0000	0.0000
	ACMDM-S-PS22+ProjFlow (Plain Masking)	0.520	<u>0.748</u>	10.335	0.0675	0.0000	0.0000	0.0000
Head	ACMDM-S-PS22+ProjFlow (Full)	<b>0.099</b>	<b>0.788</b>	<b>10.754</b>	0.0595	0.0000	0.0000	0.0000
	ACMDM-S-PS22+ProjFlow (Euclid)	<u>0.560</u>	<u>0.761</u>	<u>10.332</u>	<b>0.0547</b>	0.0000	0.0000	0.0000
	ACMDM-S-PS22+ProjFlow (w/o noise)	1.852	0.750	9.714	0.0706	0.0000	0.0000	0.0000
	ACMDM-S-PS22+ProjFlow (Plain Masking)	1.076	0.751	10.175	<u>0.0594</u>	0.0000	0.0000	0.0000
Left wrist	ACMDM-S-PS22+ProjFlow (Full)	<b>0.089</b>	<b>0.783</b>	<b>10.601</b>	<u>0.0586</u>	0.0000	0.0000	0.0000
	ACMDM-S-PS22+ProjFlow (Euclid)	0.524	0.754	<u>10.256</u>	<b>0.0583</b>	0.0000	0.0000	0.0000
	ACMDM-S-PS22+ProjFlow (w/o noise)	3.507	0.703	9.019	0.0801	0.0000	0.0000	0.0000
	ACMDM-S-PS22+ProjFlow (Plain Masking)	<u>0.506</u>	<u>0.759</u>	10.242	0.0590	0.0000	0.0000	0.0000
Right wrist	ACMDM-S-PS22+ProjFlow (Full)	<b>0.096</b>	<b>0.780</b>	<b>10.610</b>	<b>0.0584</b>	0.0000	0.0000	0.0000
	ACMDM-S-PS22+ProjFlow (Euclid)	<u>0.506</u>	0.753	<u>10.343</u>	<u>0.0591</u>	0.0000	0.0000	0.0000
	ACMDM-S-PS22+ProjFlow (w/o noise)	3.522	0.705	9.111	0.0799	0.0000	0.0000	0.0000
	ACMDM-S-PS22+ProjFlow (Plain Masking)	0.514	<u>0.759</u>	10.224	0.0594	0.0000	0.0000	0.0000
Average	ACMDM-S-PS22+ProjFlow (Full)	<b>0.097</b>	<b>0.779</b>	<b>10.651</b>	<u>0.0603</u>	0.0000	0.0000	0.0000
	ACMDM-S-PS22+ProjFlow (Euclid)	1.152	0.740	10.107	<b>0.0595</b>	0.0000	0.0000	0.0000
	ACMDM-S-PS22+ProjFlow (w/o noise)	3.429	0.707	9.308	0.0863	0.0000	0.0000	0.0000
	ACMDM-S-PS22+ProjFlow (Plain Masking)	<u>0.881</u>	<u>0.748</u>	<u>10.187</u>	0.0632	0.0000	0.0000	0.0000

trajectory-control benchmarks comparing ProjFlow with OmniControl[63] and MaskControl[63], and 2D-to-3D lifting comparisons against Sketch2Anim[69]. We refer readers to this page for a more complete visual impression of ProjFlow’s behavior.



Table 7. **Quantitative text-conditioned motion generation with spatial control signals and upper-body editing on HumanML3D.** 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. **bold face / underline** indicates the best/2<sup>nd</sup> 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
<b>Train On Pelvis</b>	MDM [57]	✓	1.792	0.673	9.131	0.1019	0.4022	0.3076	0.5959
	PriorMDM [51]	✗	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 [63]	✗	0.081	0.789	10.323	<u>0.0547</u>	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]	✗	<b>0.066</b>	0.799	10.474	<b>0.0543</b>	<b>0.0000</b>	<b>0.0000</b>	0.0093
	ACMDM-S-PS22+CtrlNet [39]	✗	<u>0.067</u>	<b>0.805</b>	<u>10.481</u>	0.0591	0.0075	0.0010	0.0100
	ACMDM-S-PS22+DNO [21]	✓	0.151	<u>0.802</u>	—	0.0610	<u>0.0027</u>	<u>0.0002</u>	<u>0.0089</u>
	<b>ACMDM-S-PS22+ProjFlow (ours)</b>	✓	0.107	0.784	<b>10.645</b>	0.0630	<b>0.0000</b>	<b>0.0000</b>	<b>0.0000</b>
<b>Pelvis</b>	OmniControl [63]	✗	0.135	0.790	10.314	<u>0.0571</u>	0.0404	0.0085	0.0367
	MotionLCM V2+CtrlNet [9]	✗	4.726	0.713	9.209	0.1162	0.1617	0.0841	0.1838
	MaskControl [45]	✗	<u>0.087</u>	0.795	10.168	<b>0.0544</b>	<u>0.0003</u>	<b>0.0000</b>	0.0114
	ACMDM-S-PS22+CtrlNet [39]	✗	<b>0.075</b>	<b>0.805</b>	<u>10.536</u>	0.0603	0.0081	0.0011	0.0134
	ACMDM-S-PS22+DNO [21]	✓	0.151	<u>0.802</u>	—	0.0610	<u>0.0027</u>	<u>0.0002</u>	<u>0.0089</u>
	<b>ACMDM-S-PS22+ProjFlow (ours)</b>	✓	0.107	0.784	<b>10.645</b>	0.0630	<b>0.0000</b>	<b>0.0000</b>	<b>0.0000</b>
<b>Left foot</b>	OmniControl [63]	✗	0.093	0.794	10.338	0.0692	0.0594	0.0094	0.0314
	MotionLCM V2+CtrlNet [9]	✗	4.810	0.706	9.158	0.1047	0.2607	0.1229	0.2304
	MaskControl [45]	✗	<u>0.074</u>	0.793	10.241	<b>0.0561</b>	<b>0.0000</b>	<b>0.0000</b>	<u>0.0066</u>
	ACMDM-S-PS22+CtrlNet [39]	✗	<b>0.063</b>	<b>0.800</b>	<u>10.542</u>	<u>0.0590</u>	0.0186	0.0034	0.0240
	ACMDM-S-PS22+DNO [21]	✓	0.147	<u>0.799</u>	—	0.0602	<u>0.0082</u>	<u>0.0003</u>	0.0133
	<b>ACMDM-S-PS22+ProjFlow (ours)</b>	✓	0.095	0.771	<b>10.644</b>	0.0609	<b>0.0000</b>	<b>0.0000</b>	<b>0.0000</b>
<b>Right foot</b>	OmniControl [63]	✗	0.137	0.798	10.241	0.0668	0.0666	0.0120	0.0334
	MotionLCM V2+CtrlNet [9]	✗	4.756	0.705	9.303	0.1026	0.2459	0.1127	0.2278
	MaskControl [45]	✗	<u>0.080</u>	0.793	10.159	<b>0.0552</b>	<b>0.0000</b>	<b>0.0000</b>	<u>0.0062</u>
	ACMDM-S-PS22+CtrlNet [39]	✗	<b>0.071</b>	<b>0.803</b>	<u>10.591</u>	<u>0.0583</u>	0.0205	0.0030	0.0251
	ACMDM-S-PS22+DNO [21]	✓	0.153	<u>0.800</u>	—	0.0597	<u>0.0086</u>	<u>0.0003</u>	0.0138
	<b>ACMDM-S-PS22+ProjFlow (ours)</b>	✓	0.096	0.770	<b>10.651</b>	0.0613	<b>0.0000</b>	<b>0.0000</b>	<b>0.0000</b>
<b>Head</b>	OmniControl [63]	✗	0.146	0.796	10.239	<u>0.0556</u>	0.0422	0.0079	0.0349
	MotionLCM V2+CtrlNet [9]	✗	4.580	0.715	9.278	0.1138	0.1971	0.0977	0.2136
	MaskControl [45]	✗	<u>0.090</u>	0.797	10.131	<b>0.0531</b>	<b>0.0000</b>	<b>0.0000</b>	<u>0.0064</u>
	ACMDM-S-PS22+CtrlNet [39]	✗	<b>0.081</b>	<b>0.805</b>	<u>10.520</u>	0.0598	0.0051	0.0009	0.0152
	ACMDM-S-PS22+DNO [21]	✓	0.138	<u>0.801</u>	—	0.0591	<u>0.0025</u>	<u>0.0002</u>	0.0084
	<b>ACMDM-S-PS22+ProjFlow (ours)</b>	✓	0.099	0.788	<b>10.754</b>	0.0595	<b>0.0000</b>	<b>0.0000</b>	<b>0.0000</b>
<b>Left wrist</b>	OmniControl [63]	✗	0.119	0.783	10.217	<u>0.0562</u>	0.0801	0.0134	0.0529
	MotionLCM V2+CtrlNet [9]	✗	4.103	0.726	9.188	0.1167	0.3965	0.1912	0.3150
	MaskControl [45]	✗	0.118	0.797	10.153	<b>0.0546</b>	<b>0.0000</b>	<b>0.0000</b>	<u>0.0044</u>
	ACMDM-S-PS22+CtrlNet [39]	✗	<b>0.065</b>	<b>0.804</b>	<u>10.480</u>	0.0604	0.0085	0.0014	0.0206
	ACMDM-S-PS22+DNO [21]	✓	0.149	<u>0.799</u>	—	0.0600	<u>0.0076</u>	<u>0.0004</u>	0.0138
	<b>ACMDM-S-PS22+ProjFlow (ours)</b>	✓	<u>0.089</u>	0.783	<b>10.601</b>	0.0586	<b>0.0000</b>	<b>0.0000</b>	<b>0.0000</b>
<b>Right wrist</b>	OmniControl [63]	✗	0.128	0.792	10.309	0.0601	0.0813	0.0127	0.0519
	MotionLCM V2+CtrlNet [9]	✗	4.051	0.725	9.242	0.1176	0.3822	0.1806	0.3079
	MaskControl [45]	✗	0.121	0.797	10.105	<b>0.0537</b>	<b>0.0000</b>	<b>0.0000</b>	<u>0.0044</u>
	ACMDM-S-PS22+CtrlNet [39]	✗	<b>0.066</b>	<b>0.802</b>	<u>10.484</u>	0.0599	0.0091	0.0016	0.0201
	ACMDM-S-PS22+DNO [21]	✓	0.143	<u>0.798</u>	—	0.0598	<u>0.0081</u>	<u>0.0004</u>	0.0142
	<b>ACMDM-S-PS22+ProjFlow (ours)</b>	✓	<u>0.096</u>	0.780	<b>10.610</b>	<u>0.0584</u>	<b>0.0000</b>	<b>0.0000</b>	<b>0.0000</b>
<b>Average</b>	OmniControl [63]	✗	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]	✗	<u>0.095</u>	0.795	10.159	<b>0.0545</b>	<u>0.0001</u>	<b>0.0000</b>	<u>0.0065</u>
	ACMDM-S-PS22+CtrlNet [39]	✗	<b>0.070</b>	<b>0.803</b>	<u>10.526</u>	<u>0.0596</u>	0.0117	0.0019	0.0197
	ACMDM-S-PS22+DNO [21]	✓	0.147	<u>0.800</u>	—	0.0600	0.0034	<u>0.0003</u>	0.0121
	<b>ACMDM-S-PS22+ProjFlow (ours)</b>	✓	0.097	0.779	<b>10.651</b>	0.0603	<b>0.0000</b>	<b>0.0000</b>	<b>0.0000</b>

Table 8. **Quantitative text-conditioned motion generation with spatial control signals and upper-body editing on HumanML3D [16].** The first section covers pelvis-only control; the middle section shows the average for all joints. The last section presents upper-body editing results. **bold face** / underline indicates the best/2<sup>nd</sup> results.

Controlling Joint	Methods	Zero-shot?	FID↓	R-Precision Top 3	Diversity→	Foot Skating Ratio↓	Traj. err.↓	Loc. err.↓	Avg. err.↓
	GT	-	0.002	0.797	9.503	-	0.000	0.000	0.000
<b>Train On Pelvis</b>	MDM [57]	✓	0.698	0.602	9.197	0.1019	40.22	30.76	59.59
	PriorMDM [51]	✗	0.475	0.583	9.156	0.0897	34.57	21.32	44.17
	GMD [20]	✓	0.576	0.665	9.206	0.1009	9.31	3.21	14.39
	OmniControl [63]	✗	0.218	0.687	9.422	<b>0.0547</b>	<u>3.87</u>	<u>0.96</u>	3.38
	MotionLCM V2 [9]	✗	0.531	0.752	9.253	—	18.87	7.69	18.97
	TLControl [59]	✗	0.271	<u>0.779</u>	<u>9.569</u>	—	<b>0.00</b>	<b>0.00</b>	1.08
	MaskControl [45]	✗	<b>0.061</b>	<b>0.809</b>	<b>9.496</b>	<b>0.0547</b>	<b>0.00</b>	<b>0.00</b>	<u>0.98</u>
	<b>ACMDM-S-PS22+ProjFlow (ours)</b>	✓	<u>0.083</u>	0.755	9.096	<u>0.0651</u>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
<b>Train On All Joints (Average)</b>	OmniControl [63]	✗	0.310	0.693	<b>9.502</b>	<u>0.0608</u>	<u>6.17</u>	<u>1.07</u>	4.04
	TLControl [59]	✗	0.256	<u>0.782</u>	9.719	—	<b>0.00</b>	<b>0.00</b>	1.11
	MaskControl [45]	✗	<u>0.083</u>	<b>0.805</b>	<u>9.395</u>	<b>0.0545</b>	<b>0.00</b>	<b>0.00</b>	<u>0.72</u>
	<b>ACMDM-S-PS22+ProjFlow (ours)</b>	✓	<b>0.074</b>	0.752	9.065	0.0624	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
	Methods	Zero-shot?	FID↓	R-Precision Top 1	R-Precision Top 2	R-Precision Top 3	Matching↓	Diversity→	—
<b>UpperBody Edit</b>	MDM [57]	✓	4.827	0.298	0.462	0.571	4.598	7.010	—
	OmniControl [63]	✗	1.213	0.374	0.550	0.656	5.228	9.258	—
	MMM [46]	✗	0.103	0.500	0.694	<u>0.798</u>	2.972	9.254	—
	MotionLCM [9]	✗	0.311	<u>0.512</u>	0.685	<u>0.798</u>	<u>2.948</u>	<u>9.736</u>	—
	MaskControl [45]	✗	<u>0.074</u>	<b>0.517</b>	<b>0.708</b>	<b>0.804</b>	<b>2.945</b>	<b>9.380</b>	—
	<b>ACMDM-S-PS22+ProjFlow (ours)</b>	✓	<b>0.051</b>	0.502	<u>0.697</u>	0.793	3.281	10.611	—