BEYOND RECOVERY: EVALUATING DYNAMICS AND PRESERVATION IN HUMAN MOTION POST-PROCESSING

Anonymous authors
Paper under double-blind review

ABSTRACT

Motion denoising and post-processing are widely used to refine recovered or generated human motion. Yet evaluations based on positional and acceleration errors can assign similar scores to outputs with different frequency content, local jerk patterns, or changes to already-clean input. We propose a dynamics-aware evaluation protocol that pairs recovery from corruption with preservation of clean motion. Alongside positional error, it adopts two established quantities: log-spectral distance (LSD) between joint-velocity power spectra, and jitter ratio, which compares jerk magnitude with a reference. An identity setting passes clean ground-truth trajectories through each post-processor using the same configuration as its recovery test. We evaluate classical filters and learned post-processors under Gaussian corruption, synthetic video-like corruption, and real estimator errors across multiple motion datasets. Methods with similar Mean Per Joint Position Error (MPJPE) differ substantially in spectral fidelity and clean-input preservation. Spectral and window-level analyses locate the affected frequencies, joints, and time intervals. Re-tuning the classical filters for recovery also increases the changes they make to clean motion. These results show why recovery gains should be evaluated alongside their cost to motion preservation.

1 INTRODUCTION

Human motion recovery (HMR) extracts human motion from monocular video, supporting motion understanding, animation, virtual avatars, and other applications (Tian et al., 2023; Shin et al., 2024; Shen et al., 2024; Zhang et al., 2025). Standalone and integrated post-processors refine the recovered trajectories, and their behavior determines which estimation errors are removed and which motion details are retained. Evaluations of motion denoising methods (Zeng et al., 2022; Wang et al., 2024a; Dong & Lee, 2025; Dang et al., 2024) commonly report mean per-joint position error (MPJPE) and acceleration error, which measure joint-position fidelity and differences in acceleration, respectively.

These two metrics capture only part of how human motion evolves over time. Positional error measures framewise agreement, while aggregate derivative errors such as acceleration error can give similar scores to trajectories with different temporal patterns (Figure 1). A post-processor can suppress both estimator fluctuations and genuine rapid movement. Persistent drift may remain after smoothing, even though the output varies less from frame to frame. Genuine motion and estimation error can occupy overlapping frequency ranges, so smoothness alone cannot distinguish them. A sequence average also does not show whether deviations are spread over the motion or concentrated in a particular joint or short interval. These cases call for measurements that distinguish pointwise geometric accuracy from the amount of movement and its distribution over time and frequency.

Input quality presents a second challenge. In motion recovery, the severity and type of error can vary across actions, viewpoints, and upstream estimators, as well as within a sequence. A post-processor may need to correct an unreliable interval while retaining the movement in an already-accurate one. Evaluation on corrupted inputs measures the ability to recover motion, but leaves open how the same method behaves when little correction is needed. Cross-dataset transfer, cross-model transfer (Zeng et al., 2022; Wang et al., 2024a), and graded corruption examine changing input conditions. The clean-input case isolates preservation by measuring how much a post-processor changes motion that is already correct.

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.

This distinction matters when developing and selecting post-processors. Filter parameters and training objectives determine how strongly an output is regularized. A temporal penalty can encourage smoother trajectories, but its effect on the retained motion depends on the input and on the other terms in the objective. Assessing that effect requires examining the resulting trajectories, including for methods trained with velocity or acceleration losses. A lower aggregate error can combine successful corrections with changes to accurate parts of the input. Evaluation should make both visible, so that the choice of a post-processor reflects the motion properties it preserves as well as the errors it removes.

We therefore propose an evaluation protocol that pairs recovery from corruption with preservation of clean motion and augments positional error with reference-relative dynamics measurements. We adopt *log-spectral distance (LSD)* to compare velocity energy across frequencies and *jitter ratio* to measure suppression or amplification of jerk relative to the reference. Comparisons with the reference give a reduction in motion its meaning: it can remove error or attenuate movement that should remain. Spectral comparisons reveal which frequencies are affected, while local jerk analysis identifies changes concentrated in particular joints or time intervals. Figure 2 illustrates why time-domain errors and spectral fidelity can rank the same post-processed outputs differently.

Preservation is tested in an *identity setting*, where clean ground-truth trajectories are passed through the post-processor. An ideal post-processor leaves these trajectories unchanged while retaining its ability to correct corrupted ones. The clean case separates changes to the reference motion from the removal of estimation error, which are mixed together in a recovery result. In use, the true estimation error is unavailable, so the post-processor must infer the need for correction from the input. We therefore use the same filter parameters or learned checkpoint in both tests, connecting the preservation cost to the configuration that produced the recovery benefit.

We interpret the paired results relative to no processing, which provides zero recovery gain and zero change to clean motion (Section 5.2). This baseline separates the contribution of the post-processor from the quality of the upstream estimate. A small final error can arise because an accurate input needed little correction, while a larger final error can accompany substantial recovery from a poorer input. For a given input condition, reporting gain together with preservation cost shows what changes when a method or its parameters are selected. Figure 3 visualizes this comparison. This comparison distinguishes configurations that improve both outcomes from those that exchange preservation for recovery. When better recovery comes with greater change to clean motion, the preferred configuration depends on how often accurate inputs occur and how costly it is to modify them. The pair supports this choice by keeping both outcomes visible.

We evaluate classical filters and learned post-processors under Gaussian corruption, synthetic video-like corruption, and real estimator errors across multiple motion datasets. The comparisons show that similar positional errors can conceal different spectral fidelity and changes to clean motion. Controlled analyses examine the sources of these differences and how they respond to method selection and input conditions. Together, the results explain why evaluating recovery alone gives an incomplete account of post-processing behavior and how paired dynamics measurements help interpret the changes made to a trajectory.

Contributions.

1. We complement positional and acceleration errors with reference-relative dynamics measurements. Log-spectral distance (LSD) describes velocity power across frequencies; jitter ratio and its local deviations describe the amount and location of jerk changes. These established quantities reveal differences hidden by aggregate errors.

2. We pair recovery with an *identity* setting that tests preservation of already-correct motion. Using the same configuration in both tests relates recovery gains to their cost on clean input, with no processing as a common reference.

3. We evaluate classical and learned post-processors across three corruption regimes. Controlled analyses explain differences in metric rankings and examine how selection, training, and upstream estimators affect the recovery–preservation balance (Section 5).

2 Related Work

Motion recovery and refinement. Video HMR extends image-based body recovery (Kanazawa et al., 2018; Kolotouros et al., 2019) with temporal and global-motion modeling (Kocabas et al., 2020; Choi et al., 2021; Shin et al., 2024; Shen et al., 2024). Classical filters (Savitzky & Golay, 1964; Casiez et al., 2012) and learned refiners (Zeng et al., 2022; Wang et al., 2024a; Dong & Lee, 2025; Dang et al., 2024) improve estimated trajectories. Velocity and acceleration losses shape these outputs during training. Our protocol measures the resulting dynamics and applies the same evaluation to classical and learned methods.

Dynamics-aware metrics. Jerk and spectral measures have precedents in motor control and motion smoothness (Flash & Hogan, 1985; Balasubramanian et al., 2012; 2015), motion prediction (NPSS) (Gopalakrishnan et al., 2019), video quality (FVMD) (Liu et al., 2024), and speech coding (LSD) (Gray & Markel, 1976; Paliwal & Atal, 1993). We adopt reference-relative jerk magnitude and velocity-spectrum LSD for post-processing evaluation. Unlike LSD, NPSS normalizes each channel's power spectrum and therefore ignores uniform amplitude scaling.

Evaluation protocols. Cross-dataset and cross-model tests (Zeng et al., 2022; Wang et al., 2024a), along with pose and tracking benchmarks (Leroy et al., 2020; Koppula et al., 2024), highlight the importance of input coverage and evaluation design. We pair recovery across three corruption regimes with clean-input identity tests, measuring whether improvements on corrupted trajectories come with changes to already-correct motion.

3 From Diagnostic to Evaluation Protocol

3.1 A Diagnostic of Modern Video-HMR Noise

We begin with the temporal structure of estimation error. For GVHMR (Shen et al., 2024), we measure framewise residuals against ground truth in the same 14-joint pelvis-aligned representation used for post-processing.

The residuals are strongly correlated in time (Figure 1, right): on 3DPW, their autocorrelation is 0.953 at a lag of one frame and still 0.282 at one second (30 frames). This persistence indicates slowly varying error that an independent framewise noise model misses. Figure 1 (left) is a constructed example of what MPJPE can miss: the two colored trajectories have the same MPJPE and the same acceleration error, but different temporal patterns.

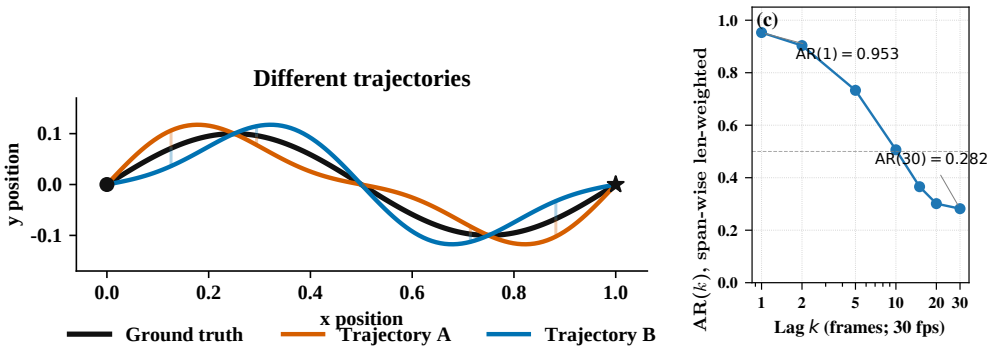


Figure 1: Temporal residuals can differ despite matching aggregate errors. Left: a conceptual example where two predictions have identical MPJPE and acceleration error but follow different temporal trajectories. Right: GVHMR residual autocorrelation on 3DPW (measured on 37 tracks from 24 videos at 30 fps, span-wise and length-weighted).

A low-pass filter may retain slowly varying error while attenuating genuine rapid motion. Comparing its output with the reference in time and frequency helps reveal these changes. The paired clean-input test then shows which changes occur even without estimation error.

3.2 METRIC DEFINITIONS

We report MPJPE for positional accuracy, jitter ratio (JR) for relative jerk magnitude, and LSD for velocity spectral fidelity. Figure 2 illustrates how spectral fidelity can disagree with time-domain errors; Section 5.1 tests this distinction across motions and post-processors.

Throughout, $x \in \mathbb{R}^{T\times J\times 3}$ is a motion sequence with J joints and T time steps at frame rate f_s. We use each dataset's native frame rate and the joint schema of the evaluated output; Appendix A.1 records these choices. Derivatives and spectra are computed within contiguous valid segments.

MPJPE. We average the Euclidean distance between predicted and reference joint positions over joints and frames. Real-estimator panels use pelvis-centered coordinates with no further per-frame alignment:

$$\mathrm{MPJPE}(x, x^{gt}) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{J}\sum_{j=1}^{J}\|x_{t,j} - x^{gt}_{t,j}\|_2 \tag{1}$$

Jitter Ratio. We define per-frame jitter as the magnitude of the third time derivative of position, averaged over joints:

$$\mathrm{jit}_t(x) \;=\; \frac{1}{J}\sum_{j}\left\|\frac{d^3x_{t,j}}{dt^3}\right\|_2, \tag{2}$$

where the derivative is the third forward finite difference on position, $f_s^3(x_{t+3} - 3x_{t+2} + 3x_{t+1} - x_t)$, in $\mathrm{m/s^3}$. We split sequences at invalid frames and obtain sequence-level jitter $\mathrm{jit}_s(x)$ by pooling segment means, weighted by their numbers of valid jerk samples.

For each sequence s in the test set S, we calculate the jitter ratio by dividing predicted jitter by ground-truth jitter. The sequence ratios r_s are then averaged within each video, subject–action group, or mirror pair, and groups are weighted equally. For a set S of equally weighted sequences, this is:

$$r \;=\; \frac{1}{|S|}\sum_{s\in S}\frac{\mathrm{jit}_s(\hat{x}_s)}{\mathrm{jit}_s(x^{\mathrm{gt}}_s)}. \tag{3}$$

For a sequence, ratios below or above one indicate lower or higher mean jerk magnitude than the reference. Opposing deviations can cancel in the dataset mean, so we also report the jitter-ratio deviation $\mathrm{JRD} = \frac{1}{|S|}\sum_{s\in S}|r_s - 1|$, whose ideal value is zero; Section 5.1 decomposes it over joints and time windows. Sequences with reference jitter at or below $10^{-8}\,\mathrm{m/s^3}$ are excluded from the ratio (none in the main tables).

Log-Spectral Distance (LSD). We compare velocity power across frequencies. For each joint–axis channel in a valid segment, we compute velocity $v_t = f_s(x_{t+1} - x_t)$ and its one-sided DFT. For a velocity sequence v in one joint-axis channel, let

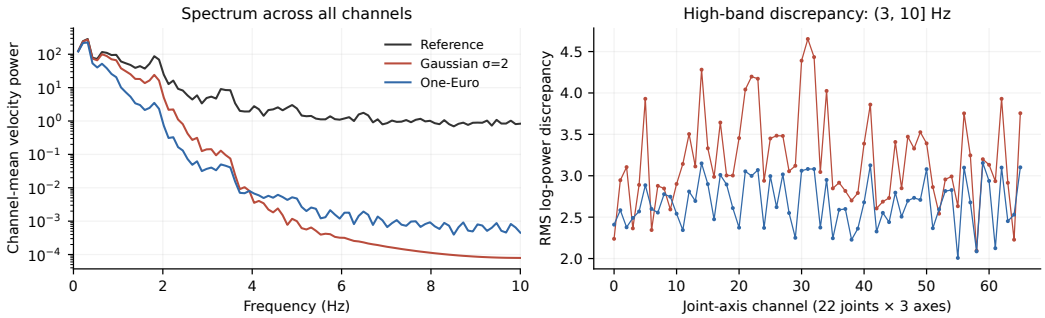
$$\widehat{v}(f) = \mathrm{DFT}(v)_f, \qquad P_v(f) = |\widehat{v}(f)|^2. \tag{4}$$

The segment-wise log-spectral distance to ground truth is

$$\mathrm{LSD}_s(v, v^{\mathrm{gt}}) = \sqrt{\frac{1}{F}\sum_{f=1}^{F}\left(\log_{10}(P_v(f) + \epsilon) - \log_{10}(P_{v^{\mathrm{gt}}}(f) + \epsilon)\right)^2}, \tag{5}$$

where F is the number of positive frequency bins and $\epsilon = 10^{-10}$ prevents undefined logarithms at zero power. We use non-DC bins with equal frequency weights and segments of at least 16 position frames. Channel scores are averaged over joints and axes, then segments are pooled by their numbers of velocity samples. Values are in $\log_{10}$ units. Both main tables use these definitions, with group-cluster intervals in Appendix B.1.

Figure 2 illustrates the distinction on a corrupted HumanML3D motion. Gaussian smoothing has lower mean per-joint velocity error (MPJVE), acceleration error, two-second windowed acceleration error, and JRD than bidirectional One-Euro, yet higher LSD. Both filters attenuate velocity power in the 3–10 Hz band; Gaussian has the larger discrepancy from the reference.



Method	MPJVE (m/s)	Accel. (m/s^2)	2 s accel.	JRD	LSD
Gaussian ($\sigma = 2$)	**0.111**	**2.050**	**2.130**	**0.791**	2.714
One-Euro	0.179	2.421	2.518	0.904	**2.358**

Figure 2: Time-domain errors and LSD rank the same outputs differently. HumanML3D sequence 013475 (189 frames, 20 fps, 22 joints). Left: velocity power averaged over 66 joint-axis channels. Right: channel-wise RMS log-power discrepancy over (3, 10] Hz. All errors use ground truth and are ideal at zero; bold/underlined values are best/second. Appendix B.5 describes example selection.

Reporting and interpretation. Comparisons use a common reference, joint schema, coordinate system, and temporal support. LSD describes non-DC velocity energy: the power spectrum discards phase, and dropping the DC bin excludes mean velocity. Frame rate, segment length, frequency weighting, and aggregation determine the quantity being compared, so we report them explicitly and examine their effects in Section 5.1. Absolute LSD values from different protocols are not pooled. Appendix A.1 specifies the full computation, and Appendix B.9 provides further temporal and spectral examples.

3.3 Paired Recovery and Identity Settings

We evaluate each post-processor in two paired settings. Let x_{gt} denote a clean reference motion, x_{corr} its corrupted version, D a post-processor, and M a metric between motion sequences.

Recovery. We apply D to estimator outputs or synthetically corrupted trajectories and measure error against the clean reference. For errors with ideal value zero, such as MPJPE, LSD, or JRD, recovery means $\mathrm{M}(D(x_{\text{corr}}), x_{\text{gt}}) < \mathrm{M}(x_{\text{corr}}, x_{\text{gt}})$.

Identity. The same post-processor receives clean ground-truth trajectories. We measure its changes to position, jerk magnitude, and velocity spectral content through

$$\Delta_{\text{clean}}[M](D) \;:=\; M\big(D(x_{\text{gt}}),\; x_{\text{gt}}\big), \tag{6}$$

where ideal preservation, $D(x_{\text{gt}}) = x_{\text{gt}}$, gives zero for MPJPE, LSD, and JRD and one for JR.

Filter parameters and learned checkpoints are held fixed across the paired tests. This connects the preservation cost to the configuration that produced the recovery benefit. Section 5.2 expresses the comparison as recovery gain and clean-input harm.

3.4 Evaluation Data

The protocol has four splits spanning three corruption regimes. They combine controlled perturbations of reference motion with outputs from real estimators, allowing us to examine both noise assumptions and post-processor behavior.

Gaussian corruption. We corrupt HumanML3D (Guo et al., 2022) motions with per-joint Gaussian noise and sparse displacement spikes, producing 4,382 test trajectories. Five severity levels span mild to extreme corruption. Appendix A.2 gives their sampling probabilities, target jitter, and realized test statistics.

Real estimator outputs. We use SPIN (Kolotouros et al., 2019) predictions on 3DPW (von Marcard et al., 2018) and FCN (Martinez et al., 2017) predictions on Human3.6M (Ionescu et al., 2014), following the pre-extracted splits used by prior post-processing work. The 37 SPIN tracks come from 24 videos; the 236 FCN tracks form 59 subject–action groups observed by four cameras. We account for shared content by resampling videos or subject–action groups when estimating uncertainty. H36M intervals are conditional on its two test subjects. Appendix A.2 records the full support and split provenance.

Synthetic video-like corruption. To examine structured errors over a larger motion cohort, we apply a manually specified corruption pipeline to HumanML3D. It adds temporally correlated noise, depth-axis bias, occlusion events, speed-dependent noise, and root drift in sequence. Three presets represent clear, average, and occluded video conditions, producing a second split of $4{,}382$ trajectories. Table 5 lists the parameters, and Appendix A.2 gives the mixture and design notes. The Gaussian and synthetic splits use the same underlying clean motions, which also supply their identity inputs.

3.5 Compared Methods

We compare Gaussian smoothing, Savitzky–Golay filtering, and bidirectional One-Euro with three learned post-processors: SmoothNet, SynSP, and PS-Mamba. Appendix Table 6 specifies their configurations. The classical filters use fixed hyperparameters across splits. Gaussian widths and Savitzky–Golay windows are set in frames, so their durations vary with the dataset's frame rate; bidirectional One-Euro averages aligned forward and reversed passes. Section 5.2 separately examines selection on a shared validation cohort and recovery objective.

Learned post-processors are trained for each split, with one checkpoint used in both recovery and identity tests. Appendix Table 7 records the available losses, optimization settings, and checkpoint selection rules. To examine the effect of the objective under controlled training, Section 5.3 evaluates six combinations of acceleration and spectral loss weights with ten paired SmoothNet seeds.

4 Results Analysis

Tables 1 and 2 report the same post-processors under recovery and identity on all four splits. We first examine where positional and dynamics measurements give different accounts of the outputs, then compare recovery with the changes made to clean input. Both tables use the definitions of Section 3.2; Appendix B.1 adds JRD and group-cluster intervals.

Similar positional errors can hide different dynamics. In the recovery setting (Table 1), Gaussian smoothing has the lowest MPJPE on 3DPW-SPIN but the largest LSD among comparable post-processors. All five methods lie within 86.4–88.7 mm MPJPE, yet some suppress jerk and others amplify it relative to the reference. The difference is visible in the other regimes as well: SynSP reduces positional error on H36M-FCN and HumanML3D-Synthetic while suppressing jerk. Positional accuracy alone leaves these temporal effects unresolved. Appendix Figure 8 illustrates two outputs with the same displacement from a raw estimator trajectory but different spectral discrepancies.

Clean-input tests reveal preservation costs. In the identity setting (Table 2), all three classical filters suppress clean-motion jerk. Even Savitzky–Golay, which has the smallest positional change on every split, lowers mean jerk to less than one-third of the reference on H36M. The learned methods also alter clean dynamics: SynSP suppresses jerk, whereas PS-Mamba amplifies it. SmoothNet's behavior depends on the split. On HumanML3D-Gauss it has lower JRD and LSD than any classical filter; on clean 3DPW it moves joints by 20.9 mm, more than any filter. The identity test exposes this variation in preservation alongside the recovery results (Section 5.2).

The filter ranking changes with the corruption regime. The synthetic inputs have both larger positional error than the Gaussian inputs (65.90 versus 7.43 mm) and a different temporal structure. Every method reaches lower MPJPE on the Gaussian split, including learned methods trained for each regime. The filter ordering changes too: Savgol(9,3) has the lowest MPJPE of the three filters on the Gaussian split and the highest on the synthetic split. A filter choice made on Gaussian corruption

Table 1: Recovery setting (MPJPE in mm). Bold/underlined: best/second among comparable post-processors, minimizing MPJPE, LSD, and $|\mathrm{JR}-1|$. No processing is unranked. Means weight 3DPW videos, H36M subject–action groups, and HumanML3D mirror pairs equally. †: 3DPW PS-Mamba uses 14 rather than 17 joints and is unranked. All 4,382 HumanML3D sequences are included; 12 lack minimum LSD support.

Method	3DPW-SPIN			H36M-FCN			HumanML3D-Gauss			HumanML3D-Synthetic		
	MPJPE	JR	LSD	MPJPE	JR	LSD	MPJPE	JR	LSD	MPJPE	JR	LSD
No processing	88.2	7.188	1.300	54.1	34.054	2.221	7.4	16.787	1.469	65.9	15.766	2.060
Gaussian (σ=2)	**86.4**	0.196	3.573	53.1	<u>0.676</u>	2.353	11.4	0.420	1.966	61.0	<u>1.371</u>	1.459
Savgol (9,3)	<u>87.0</u>	**1.125**	<u>1.115</u>	53.4	5.069	<u>1.223</u>	**5.9**	2.579	<u>1.055</u>	64.7	4.087	1.463
1Euro-bidir	87.5	<u>0.701</u>	**1.087**	55.6	2.051	**0.977**	25.2	2.370	1.195	63.2	2.498	1.344
SmoothNet	88.0	1.323	1.406	<u>52.6</u>	**1.128**	1.264	<u>6.5</u>	<u>1.430</u>	**0.970**	**48.9**	1.546	**1.255**
SynSP	88.7	0.178	2.408	**52.4**	0.674	1.341	29.7	**1.050**	1.224	<u>60.4</u>	**0.779**	<u>1.302</u>
PS-Mamba†	108.8	10.245	1.304	56.4	19.289	2.367	87.2	22.942	2.108	90.0	16.144	1.991

Table 2: Identity setting; MPJPE and LSD are ideal at 0, JR at 1. Ranking notation, configurations, cohorts, and aggregation follow Table 1. SmoothNet zero-pads the 462 HumanML3D sequences shorter than 64 frames; Appendix A.3 records inference and training settings.

Method	3DPW-SPIN			H36M-FCN			HumanML3D-Gauss			HumanML3D-Synthetic		
	MPJPE	JR	LSD	MPJPE	JR	LSD	MPJPE	JR	LSD	MPJPE	JR	LSD
No processing	0.0	1.000	0.000	0.0	1.000	0.000	0.0	1.000	0.000	0.0	1.000	0.000
Gaussian (σ=2)	<u>5.3</u>	0.091	3.945	<u>1.3</u>	0.144	2.998	10.1	0.282	2.242	<u>10.1</u>	0.282	2.242
Savgol (9,3)	**2.4**	<u>0.233</u>	<u>1.531</u>	**0.4**	<u>0.282</u>	<u>1.354</u>	**3.2**	0.476	<u>1.015</u>	**3.2**	<u>0.476</u>	**1.015**
1Euro-bidir	19.5	0.106	1.696	14.2	0.074	2.004	24.5	0.222	1.279	24.5	0.222	1.279
SmoothNet	20.9	**1.106**	**1.406**	4.3	**0.695**	**1.201**	<u>4.0</u>	**1.045**	**0.919**	18.7	**0.694**	<u>1.095</u>
SynSP	68.9	0.061	2.754	29.4	0.173	1.827	29.4	<u>0.535</u>	1.219	47.0	0.327	1.381
PS-Mamba†	77.1	10.737	1.308	28.1	20.470	2.354	87.1	22.851	2.101	77.6	15.072	1.911

therefore need not carry over to the structured errors in the synthetic split. Section 5.2 examines this change directly by re-selecting filter parameters.

5 Additional Analyses

We next examine what drives the metric disagreements, how parameter selection changes the recovery–preservation balance, and how the results depend on training and upstream estimators.

5.1 Spectral and localized diagnostic value

The disagreement in Figure 2 recurs across the 1,960 HumanML3D-Gauss mirror groups with at least 64 frames. We compare LSD with the four time-domain errors shown in the figure across the corrupted input and eight post-processed outputs. Of their 36 pairs, 16 have opposite rankings under LSD and at least one time-domain error after Holm correction; four disagree with all four errors.

The distinction also appears when predicting LSD from a broader set of time-domain summaries. Cross-validation holding out both source motions and method families gives $R^2 = 0.239$ (95% CI $[0.220, 0.260]$), leaving much of the variation in LSD unexplained.

Velocity and acceleration errors aggregate absolute differences in the time domain, whereas LSD compares relative power differences through the logarithm. A frequency band with little absolute energy can therefore contribute appreciably when a filter removes most of its power. Inspecting the spectrum shows whether a discrepancy comes from suppressing reference motion or retaining excess variation. In the controlled perturbations, LSD also responds to uniform amplitude changes that normalized NPSS ignores; both spectral measures are insensitive to phase (Appendix B.7).

LSD rankings also depend on frame rate, temporal support, and frequency weighting. For comparisons on new data, temporal support should be stated in seconds and the scored frequency range fixed in advance, with heterogeneous frame rates or durations analyzed separately. Band weights define the spectral target; checking their effect on rankings helps assess that protocol choice. Appendix B.5 reports these sensitivities, together with the correlations and paired tests.

Local jerk analysis shows where temporal changes occur. For the SmoothNet baseline, 92.6% of groups contain both joint–window regions with excess jerk and regions with suppressed jerk. Pooling jerk magnitudes before taking a ratio allows excess jerk in one region to offset suppression elsewhere. Computing deviations within each joint–window region exposes both changes. Finer aggregation increases mean JRD while leaving the overall method ordering largely stable. The local breakdown therefore identifies affected joints and intervals even when it changes little about the ranking (Appendix B.6).

5.2 Recovery gain against clean-input harm

For an error d with ideal zero, we compare a configuration's recovery gain with its clean-input harm:

$$G_d = \mathbb{E}[d(x_{\text{corr}}, x_{\text{gt}}) - d(D(x_{\text{corr}}), x_{\text{gt}})],$$
$$H_d = \mathbb{E}[d(D(x_{\text{gt}}), x_{\text{gt}})].$$

No processing gives $(G_d, H_d) = (0, 0)$. For a clean-input fraction π and relative harm cost λ, expected improvement is $U_d = (1 - \pi)G_d - \lambda\pi H_d$. When these application-specific quantities are unknown, we report the pair (G_d, H_d) and its uncertainty. Selecting only for recovery corresponds to $\pi = 0$.

We first select the three classical filter families and SmoothNet on the same Gaussian-split validation cohort and recovery-MPJPE objective, with family-specific search and training budgets. All four selected configurations improve recovery and alter clean input (Appendix Table 9). Sequence-length stratification explains much of SmoothNet's preservation cost: zero-padding affects its short inputs, whereas on sequences of at least 64 frames it achieves the largest recovery gain (Appendix B.2).

We then re-select the classical filters on synthetic validation data using the same search grids and objective. Figure 3 shows a consistent shift: better recovery is accompanied by greater clean-input harm. Paired recording-cluster intervals exclude zero for both changes in every family (Appendix Table 11). Recovery-based selection thus changes the preservation behavior of the chosen filter as well as its performance on the target corruption.

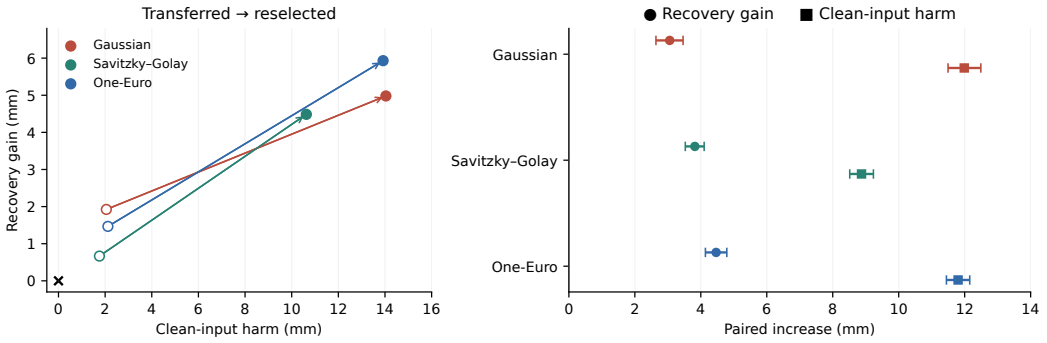


Figure 3: Recovery gain and clean-input harm under filter re-selection on HumanML3D-Synthetic. Left: arrows connect settings selected on Gaussian-corrupted data (open markers) to settings re-selected on synthetic validation data (filled markers); no processing is at $(0, 0)$. Right: paired increments with 95% recording-cluster bootstrap intervals. All three families improve recovery while increasing clean-input harm; Appendix Table 11 gives the full results.

A move up and to the left improves both outcomes. When gain and harm both increase, U_d gives the preference for a specified input mixture and preservation cost. Recording identity performance during validation would make that choice explicit.

5.3 Training and estimator dependence

For learned post-processors, the choice of training objective can also change the output dynamics. In a SmoothNet study with six loss settings and ten paired seeds, the effect of a spectral term on LSD depends on the acceleration-loss weight. At the lower acceleration weight, increasing the spectral weight raises LSD; at the higher weight, the estimated LSD decrease has an interval that includes zero. Joint seed–video intervals for the interaction exclude zero, while MPJPE rises slightly at both acceleration weights. The loss combinations therefore need to be evaluated jointly for spectral fidelity and positional accuracy (Appendix B.8).

The upstream estimator matters too. On the same 3DPW videos, ScoreHMR (Stathopoulos et al., 2024) improves both position and dynamics from its HMR2 (Goel et al., 2023) initialization, while a SmoothNet checkpoint transferred from SPIN reduces VIBE's (Kocabas et al., 2020) jitter but raises its MPJPE. On identical HMR2 inputs, the classical filters also exchange their positional and spectral rankings. Because the input and scored frames are shared within this comparison, the disagreement comes from how the post-processors modify the same estimator output.

The EMDB (Kaufmann et al., 2023) analyses extend the evaluation to another recording cohort and several estimators, including the camera and world stages of DuoMo (Wang et al., 2026). Appendix B.8 reports the estimator-specific effects and the auxiliary inputs used by each pipeline.

External outcomes. Two further probes examine post-processing through pressure-sensor foot-contact labels and text–motion retrieval. Contact is defined by pressure measurements, allowing foot motion during contact to be scored independently of a motion-based contact detector. Gaussian filtering reduces pressure-conditioned foot slip on corrupted input but increases it on clean input, so the recovery–preservation trade-off also appears in a physical outcome. On corrupted input, the improvement in foot slip accompanies a rise in positional error, illustrating how spatial fidelity and contact behavior can favor different outputs. Text–motion retrieval improves under some post-processors and degrades under others. Appendix B.10 gives the evaluation setup and results for both probes.

6 Limitations

The protocol measures fidelity to a reference motion. In the identity setting, it counts any departure from the annotation, including changes intended to improve physical plausibility. Agreement with human judgments remains untested.

The synthetic corruptions are manually specified, and real estimators can produce additional error structure. HumanML3D's training and test splits share source recordings, including overlapping crops (Appendix A.2); group-level resampling does not remove this dependence.

7 Conclusion

A post-processor can reduce positional error while altering the dynamics of the reference motion. A configuration selected to recover corrupted trajectories can also introduce error on clean ones. Our protocol makes both effects visible by pairing recovery and identity tests with positional error, reference-relative jerk, and velocity-spectrum LSD.

The analyses explain the resulting differences in terms of frequency content, local jerk changes, and the preservation cost of parameter selection. They also show why the upstream estimator and training objective belong in the interpretation of a post-processing result. Future work can extend the temporal analysis with timing-aware measurements and test its relationship to human judgments. Studies of training objectives can use the same paired tests to examine whether better recovery retains already-correct motion. Reporting recovery and preservation together provides a direct way to evaluate those choices.