COLOURING MOTION: FORWARD-PROCESS DESIGN FOR MULTI-AGENT TRAJECTORY DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models are a popular choice for multi-agent trajectory generation. Typically, the noising process uses stochastic white noise, even though real trajectories evolve smoothly over time, with the position at one time step affecting the next. We argue that this mismatch presents an extra burden to the denoising process. In this paper, we replace white noise with a range of temporally correlated noise fields that can be tuned to suit the underlying process being modelled. Sport provides us with large standardised datasets to work with. Compared to the same model trained with white noise, correlated noise models reduced averaged scene error by 5.1% on association football and 2.3% on American football. Our best model beats the current state-of-the-art on a public sport trajectory generation benchmark, improving averaged and best-of-20 scene error on Bundesliga football by 9.6% and 5.9%. We further explore the properties that make temporal noise fields effective, showing that slower-varying noise helps performance. The training objective is equally important. Whitening the loss has a negative effect on the results compared to plain squared loss. We also demonstrate that at 25 Hz, varying the noise by agent type makes the ball more realistic. These results establish the importance of tuning the forward process in multi-agent diffusion models.

1 INTRODUCTION

Diffusion models generate new data by corrupting training data with noise, in the forward process and teaching a model to reverse that noising in the reverse process (Ho et al., 2020). The noise is almost always white: Gaussian, and independent across every coordinate. This is useful in images which have no time axis and contain detail at every scale. However, this is not the case for human trajectories.

In spite of this, state-of-the-art multi-agent trajectory systems, such as those used in sport, use white noise in the forward process (Capellera et al., 2025; 2026). However, human trajectories align better with structured noise, which matches motions' temporally correlated nature (neighbouring frames in a motion time-series have a strong relationship). In addition, the motions do not tend to contain very high frequencies; unhelpful frequencies can be filtered out in our proposed method.

Multi-agent systems, like those found in sport, are composed of multiple agent-types and associated behaviours. Sport is a good test bed for this type of research as it provides lots of standardised data where players and a ball coexist. In football, for example, the players move smoothly (under continuous control). In contrast, when a ball is struck, there is a spike in acceleration before it naturally decays. Hence, sharing a forward process across agents is not always optimal.

To assess the role of dependence in noise, we sample the noise from a Gaussian distribution that is correlated across frames, which we call a *noise field*. To study its effect, we carried out an ablation study across different correlations in the noise field and compared the effect to independent white noise. We also assessed the role of changing the noise field by agent type. To do this, we carried out a secondary ablation study where we compared the effect of homogeneous and heterogeneous forward process noise.

Contributions

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.

- *Establishes a new state-of-the-art for football trajectory generation.* On the public Bundesliga JointDiff benchmark (Capellera et al., 2026), our design improves upon averaged scene error (-9.6% over 20 samples, -5.9% on the best of 20 samples) and end of scene error (-9.9% over 20 samples, -6.3% on the best of 20 samples). This also ports to NFL (-7.0% averaged, -5.8% on the best of 20) and NBA (-6.7% averaged, -4.7% on the best of 20).
- *Demonstrates how to match noise to data.* By performing ablation on types of noise, we show that slower-varying noise (more similar to human movement) reduces displacement error for this use case.
- *Establishes per-agent-type forward processes as a design choice in multi-agent diffusion.* We use different noise for different agent-types and show that this decreases jitter in the ball, making it more realistic.

2 Related Work

Previous work has pointed to many ways to shape the forward process, as a means of optimisation. Hoogeboom & Salimans (2023) use blurring as their noising mechanism, and Rissanen et al. (2023) reach the same construction through heat dissipation. Bansal et al. (2023) and Daras et al. (2023) went further and replace the corruption with arbitrary degradations. Falck et al. (2025) explain what these have in common: the forward process decides which frequencies are corrupted first, and so decides the order in which the model learns to undo them.

For cases when data points are related, such as videos or motion, several studies have found dependence in the corrupting noise to be useful. Voleti et al. (2022) generalise the forward process to non-isotropic Gaussian noise. Lee et al. (2022) use data-dependent diagonal covariance and Biloš et al. (2023) use correlated noise for a continuous time series. In images and video the same idea appears as a spatial or temporal noise prior (Huang et al., 2024; Ge et al., 2023; Chang et al., 2024). As far as we know, no work varies the forward process by agent-type in a multi-agent system.

Sport is often used to develop multi-agent systems, and gives us an opportunity to explore the optimisation of such systems. U2Diff and JointDiff generate whole scenes of basketball, American football, and association football (Capellera et al., 2025; 2026), and PlayBest plans basketball behaviour with a learned value function (Chen et al., 2024).

To generate whole sports scenes, the ball's and players' trajectories must be generated. However, they move in fundamentally different ways. Previous work, e.g. JointDiff (Capellera et al., 2026), learned the difference implicitly. Kim et al. (2023) models the ball's position separately from the position of the players. In this work, we used a single model but noised each element differently; we found that this increased ball realism.

In post hoc evaluation, displacement error is generally used to assess model performance. This measures the difference between generated and observed trajectories, reported over K samples (Alahi et al., 2016; Gupta et al., 2018; Capellera et al., 2026). However, displacement error says nothing about whether generated motions are physically plausible. Yeh et al. (2019) inspected distributional metrics (informally through box plots) to compare behaviours such as accelerations between the generated data and ground-truth data. Inspired by this work, we have formulated several realism metrics (Section 3.3), which we use in this work.

3 Methods

3.1 Problem Formulation

A scene is a fixed-length window $x_0 \in \mathbb{R}^{T \times N \times [x,y]}$ from tracking data with planar coordinates for N agents over T frames. On the Bundesliga benchmark, $T = 40$ and $N = 23$ agents comprising 11 players each for two competing teams and a single ball. The trajectory generation task is to produce the last 30 frames given the first 10 frames as context. At the given sampling rate of $6.25\,\text{Hz}$, this results in a model that observes $1.6\,\text{s}$ and generates $4.8\,\text{s}$.

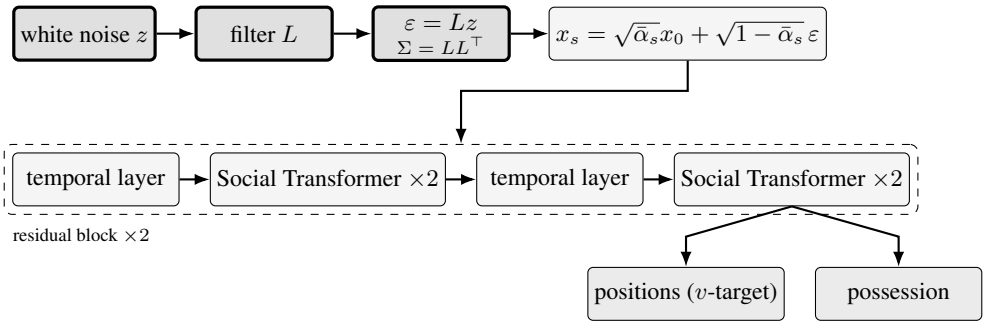

Figure 1: **The architecture of the model used in this paper:** White noise z is passed through a fixed linear filter L to give the corrupting noise ε, which has covariance $\Sigma = LL^\top$; white noise is the case $L = I$. The denoiser and both output heads are those of Capellera et al. (2026) and are held fixed throughout.

The task is probabilistic rather than deterministic. Generative models of this type produce K samples for each scene, each one a different continuation of the same observed frames. We followed the benchmark and used $K = 20$.

The forward process takes a scene x_0 and adds noise ε to it, increasing over a schedule of S steps. This is indexed by the diffusion step s,

$$x_s = \sqrt{\bar{\alpha}_s}\, x_0 + \sqrt{1-\bar{\alpha}_s}\, \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \Sigma), \tag{1}$$

where $\bar{\alpha}_s$ (signal remaining) decreases from 1 to near 0 over the S steps, so x_S is almost pure noise. The reverse process starts from noise and removes it step by step.

The noise ε is drawn from a zero-mean Gaussian distribution with covariance Σ. Standard practice sets $\Sigma = I$, which makes ε white. To correlate noise, we varied Σ.

3.2 Design

Base model. We adopt the architecture of JointDiff (Capellera et al., 2026), shown in Figure 1, and make two changes: 1) we use correlated noise instead of white noise in the forward (noising) process, and 2) we predict the velocity v (a blend of noise and signal) instead of the noise ε during the reverse (denoising) process.

Prediction target. JointDiff is trained to predict the noise ε and remove it. We instead predict the velocity target $v = \sqrt{\bar{\alpha}_s}\,\varepsilon - \sqrt{1-\bar{\alpha}_s}\,x_0$ which is a linear combination of the noise ε and the clean sample x_0 (Salimans & Ho, 2022). We do this because predicting ε becomes uninstructive at high noise levels (late on in the schedule). We report this change separately from the forward process throughout, so that neither is credited with the other's effect.

Noise fields. A *noise field* is the noise ε of Equation 1 sampled from a Gaussian distribution. This is white when the covariance Σ is the identity. We build every non-white field as

$$\varepsilon = Lz, \qquad z \sim \mathcal{N}(0, I), \qquad \Sigma = LL^\top, \tag{2}$$

where L is a fixed linear filter that acts along the time axis. In our formulation of a noise field, we used a linear filter to prevent the noise from becoming non-Gaussian, allowing us to keep the process closed-form. We scaled L so that each coordinate of ε has unit variance, which keeps the $\sqrt{1-\bar{\alpha}_s}$ scaling in Equation 1 calibrated and makes fields comparable with each other and with white noise.

We used seven fields across two levels with their parameters listed in Table 3:

Spectrum-level

- *White.* $L = I$ (the baseline).
- *Low-pass.* Gaussian kernel, at two widths, $\sigma = 0.5$ and $\sigma = 0.95$ frames.

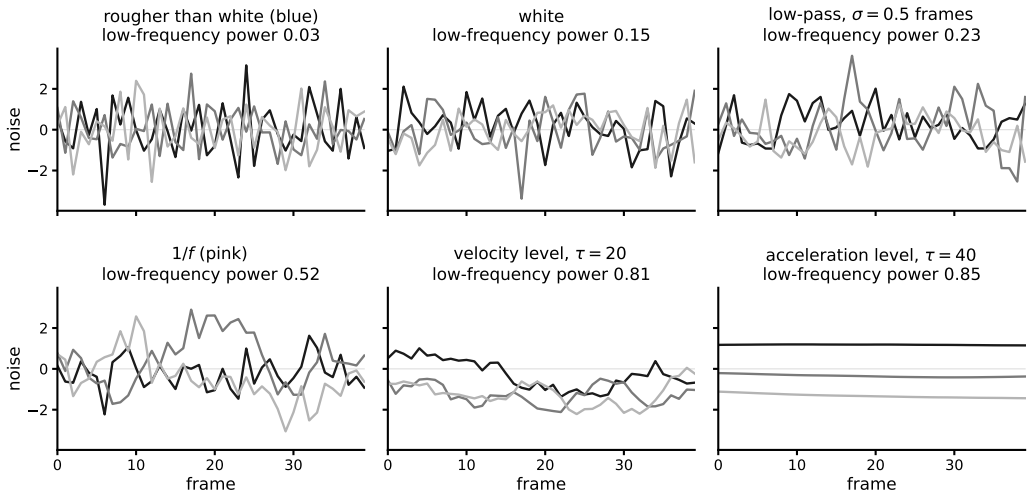

Figure 2: A visualisation of the noise fields. Each panel shows three independent samples of the noise applied to one player's x coordinate over the benchmark's 40 frames, on a common vertical scale, with the panels ordered by low-frequency power as Table 3 is. The second low-pass width is omitted for space and sits between low-pass $\sigma = 0.5$ and $1/f$.

- $1/f$ *(pink).* Fourier synthesis with $\beta = 1$.
- *Rougher than white (blue).* Fourier synthesis with $\beta = -1$, so power grows with frequency. This is the control in the opposite direction.

Derivative-level We applied a linear recursion along frames to white noise z:

$$\text{velocity level:} \quad \varepsilon_t = \rho\, \varepsilon_{t-1} + z_t, \tag{3}$$

$$\text{acceleration level:} \quad u_t = \rho\, u_{t-1} + z_t, \qquad \varepsilon_t = \varepsilon_{t-1} + u_t, \tag{4}$$

with $\rho = \exp(-1/\tau)$ and τ a damping time measured in frames. Equation 3 is an AR(1) process, the discrete-time Ornstein–Uhlenbeck process, and corresponds to adding white noise to each agent's velocity and accumulating it into position. Equation 4 adds it to acceleration and accumulates twice. In both cases the corruption is still applied to the positions in Equation 1. A larger value of τ means a longer-lived perturbation.

- *Velocity level.* Equation 3 with $\tau = 20$ frames.
- *Acceleration level.* Equation 4 with $\tau = 40$ frames.

The only exception to this field formulation is the field we designed for the ball specifically. We adapted the process to mimic the ball's sparse impulses. In Equation 4, the z term was multiplied by $k = 25$ on a random fraction $p = 0.056$ of frames and left alone elsewhere. This was then accumulated twice with $\tau = 12$ frames. Scaling the noise by a random amplitude makes this field a Gaussian scale mixture rather than a Gaussian. Appendix G evaluates the two steps of its construction separately.

A field's properties. We use two summary statistics, both computed from 512 samples of the noise field. *Low-frequency power* is the share of the field's total variance carried by frequencies below 0.1 cycles per frame. White noise has 0.15 by this measure and a field is nearly smooth as it approaches 1 (Figure 2, last panel). The threshold is in cycles per frame, not cycles per second, so the same filter has different low-frequency power at different sampling rates; ten frames is 1.6 s at 6.25 Hz and 0.4 s at 25 Hz. The *participation ratio* is $(\sum_i \lambda_i)^2 / \sum_i \lambda_i^2$, where λ_i are the eigenvalues of the field's $T \times T$ covariance matrix. This counts how many independent temporal patterns the field contains. White noise over 40 frames gives 39.6, close to the maximum of T; a field whose every sample is the same curve with a random amplitude gives 1.

Fields per-agent-type. Equation 2 allows a different filter for each agent type, because L is a matrix, so it can have one axis for the players and one for the ball. We call a model *agent-type-specific* when the two types get different filters and *shared* when one filter is applied to every agent.

Objective. For noise $\varepsilon \sim \mathcal{N}(0, \Sigma)$, the variational bound (the standard likelihood-based objective) weights the noise-prediction error by Σ^{-1}, i.e. measures it in units of standard deviation, of the noise, in each direction. Because $\Sigma = LL^\top$, weighting the error by Σ^{-1} is the same as taking plain squared error on $L^{-1}(\varepsilon - \varepsilon_\theta)$. L^{-1} is the transformation that turns the field's correlated noise back into white noise, so we call this *whitened loss*. Because our model predicts v, we whiten its v error; this differs from whitening the ε error only by a scale factor at each diffusion step. Lee et al. (2022) train with whitened loss, and Biloš et al. (2023) used the equivalently standardised noise. However, for this study, using whitened loss gave worse results than the squared loss; see Appendix D for details. Hence, we use squared loss throughout.

3.3 Evaluation Framework

As discussed above, we use two types of metrics in our evaluation: displacement error, and a range of ball acceleration metrics.

Displacement error. For each sample, the following is calculated: the scene average displacement error (SADE), this is the mean Euclidean distance between generated and ground-truth positions, averaged over all agents and all generated frames. The scene final displacement error (SFDE) is the same quantity calculated at the last generated frame. Both are reported in metres, except on NFL where the benchmark uses yards. Following the benchmark, we summarise the $K = 20$ samples of a scene in two ways (Capellera et al., 2026). 1) we take the *min* value of the set of 20 errors. 2) we take the *avg* value of the set of 20 errors. The *min* gives a value for how close the model came to the ground truth. The *avg* gives a measure of consistency. Percentage changes are computed from unrounded means.

Error by agent type. SADE gives every agent equal weight, so a change confined to the ball only moves the SADE by about $1/23$ of its true size. We therefore also report the ball's average displacement error (avgBADE), computed for the ball alone, and the players' displacement error (avgPADE). avgPADE is recovered from the identity $\text{avgPADE} = (N \cdot \text{avgSADE} - \text{avgBADE})/(N-1)$.

Ball acceleration. A model can score well on displacement error while producing a ball that jitters from frame to frame. We measure the magnitude of the generated ball's frame-to-frame acceleration, $|\Delta^2 x|$, in metres per frame squared, pooled over the generated frames of every sample. We summarise its distribution by the median (the 50th percentile), the 99th percentile (q_{99}), and the 1-Wasserstein distance W_1 between the model's distribution of $\log_{10}$ magnitudes and ground truth's. The median describes the ball at rest, the 99th percentile describes it when struck, and W_1 describes the whole distribution, with 0.30 corresponding to a factor of two in magnitude.

We also report the ratio q_{99}/q_{50}, which we call *burstiness*. We report this because the distributional metrics can be optimised via smoothing alone, but this metric cannot (Appendix H). These metrics are compared to the ground truth to assess realism.

Uncertainty. We compared models on the same samples. Where comparisons were repeated across training seeds, we give the range or the standard deviation over seeds. Because of compute constraints, we could not always repeat seeds. When one seed was used, we give a 95% interval from a bootstrap over scenes (resampling $10{,}000$ times and recomputing the paired difference).

3.4 Protocol

We ran four experiments: 1) We produced a white noise baseline for comparison with our best models (Section 4.1). We included vanilla JointDiff as ε-prediction with white noise shown in ablation tables for reference. 2) We examined the effect of the seven noise fields, defined in Section 3.2 to isolate the effect of the noise field (Section 4.2). 3) We analysed the effect of sampling rate on model

performance with correlated noise (Section 4.3). We did this by downsampling our 25 Hz tracking data to 6.25 Hz and trained models on both. For each sampling rate, we trained a model with a low-pass field and a white noise control. 4) We analysed the effect of agent-type-specific forward processes compared to shared forward processes (Section 4.4). We did this by holding the players' field fixed (as low-pass, $\sigma = 0.5$) and changing the ball's noise field. We gave the ball its own noise field as outlined in Section 3.2 and compared it to a model in which the ball had a low-pass field that matched the players. We also compared it with two white noise controls. In the first, the players kept the low-pass field, and the ball was given white noise. In the second, both agent types were given white noise. We also trained a model in which the players shared the ball's acceleration-level field. We conducted this at 25 Hz. We then repeated the central comparison (shared low-pass compared to the bespoke ball field) on the 6.25 Hz Bundesliga benchmark.

Supplementary material is shown in the Appendices. Supporting experiments are as follows: the effect of the number of sampling steps (Appendix C), sample diversity (Appendix B), the training objective (Appendix D), and the components of the ball's field (Appendix G).

4 Results

4.1 Benchmark

In this section we detail how we consistently beat the benchmark scores across several sports.

Our best model (utilising a velocity-level noise field) had a lower displacement error than every published method on minSADE, avgSADE, minSFDE, and avgSFDE for the Bundesliga benchmark. Table 1 shows these figures and Appendix J shows a visual output.

The table also separates the respective effects of our two changes. Differences in our pipeline moved the error set by the published row by between -2.8% and $+0.8\%$ across the four metrics. Switching to v-prediction worsened best-of-20 error across the samples but improved averaged error, moving the four metrics by between -3.7% and $+4.0\%$. The noise field improved all four by between 3.7% and 8.9%.

Table 1: Bundesliga future generation, the forecasting task of the JointDiff benchmark, metres, lower is better. Baseline rows as published by Capellera et al. (2026), unguided setting. The full baseline table is in Appendix I.

method	minSADE	avgSADE	minSFDE	avgSFDE
U2Diff (Capellera et al., 2025)	2.69	4.21	5.46	9.44
JointDiff (Capellera et al., 2026)	2.47	3.66	5.02	8.29
ε-pred, white noise	2.490 ± 0.016	3.611 ± 0.010	4.955 ± 0.034	8.057 ± 0.024
ours, white noise	2.552 ± 0.017	3.485 ± 0.029	5.154 ± 0.037	7.761 ± 0.070
ours, low-pass noise	2.511 ± 0.020	3.454 ± 0.016	5.111 ± 0.041	7.745 ± 0.045
ours, velocity-level	$\mathbf{2.324 \pm 0.033}$	$\mathbf{3.308 \pm 0.016}$	$\mathbf{4.705 \pm 0.067}$	$\mathbf{7.471 \pm 0.039}$

Our best model also reduced displacement error across all four metrics for NFL and NBA, as shown in Table 2. On basketball, however, the change was driven by the parameterisation and the pipeline. Our white noise model already improved on every published method, and changing to a correlated noise field lowered average scene error by at most a further 0.1% (low-pass, 1.866 against 1.868). This small improvement is negligible.

4.2 Ablation

The investigation into the effect of noise fields showed that their low-frequency power (fraction of field power below 0.1 cycles per frame) is inversely correlated with averaged scene error (Spearman's $\rho = -0.94$; see Appendix E). Models whose noise field had more low-frequency power performed the trajectory generation task better. The lowest error had a plateau, with pink and velocity level noise fields nearly indistinguishable. Although pink had a marginally lower avgSADE than the velocity model here, the result was not consistent over multiple seeds, which is why the velocity-level model is our best model in Section 4.1.

Table 2: American football (NFL, yards) and basketball (NBA, metres) future generation, lower is better, best in bold. Baseline rows as published by Capellera et al. (2026), unguided setting. The full baseline table is in Appendix I.

	American football (NFL)				basketball (NBA)			
	SADE		SFDE		SADE		SFDE	
method	min	avg	min	avg	min	avg	min	avg
MoFlow	2.33	4.02	5.51	9.98	1.52	2.42	2.73	4.64
U2Diff	2.59	3.74	5.97	9.02	1.48	2.12	2.68	4.14
JointDiff	2.36	3.40	5.53	8.40	1.39	2.01	2.53	3.95
ε-pred, white noise	2.294	3.322	5.433	8.263	1.334	1.920	2.454	3.814
ours, white noise	2.290	3.235	5.417	8.041	1.322	1.868	2.435	3.712
ours, low-pass noise	2.236	3.188	5.322	7.948	**1.316**	**1.866**	2.430	3.713
ours, velocity-level	**2.196**	**3.161**	**5.218**	**7.911**	1.325	1.876	**2.429**	**3.704**

Acceleration level noise did not fit this trend. It had the most low frequency power, but also the highest error. This error was extremely high (38.8 m on 105×68 m pitches), suggesting that the model was not able to perform the task at all. This is because it perturbs trajectories by essentially a random offset rather than time-varying noise (shown in Figure 2 and by its participation ratio of 1.0 in Table 3). This suggests an inability for the acceleration level noise field to actually noise a trajectory in a way that a denoiser can learn from, rather than a reversal of the low frequency power trend in the noise fields.

Table 3: Noise fields on the Bundesliga benchmark, ordered by low-frequency power. This sweep used one seed so numbers are not identical to multi-seed equivalents elsewhere.

noise field	low-frequency power	participation ratio	avgSADE
rougher than white (blue)	0.03	29.9	4.061
white	0.15	39.6	3.489
low-pass, $\sigma = 0.5$ frames	0.23	35.0	3.470
low-pass, $\sigma = 0.95$ frames	0.49	17.1	3.377
$1/f$ (pink)	0.52	8.3	3.318
velocity level	0.81	2.7	3.329
acceleration level	0.85	1.0	38.803

4.3 Sampling rate

At 25 Hz, the low-pass field reduced averaged scene error by 10.5% against white noise. On the same data downsampled to 6.25 Hz, it reduced it by 1.8% (Table 4). Both intervals exclude zero.

The reduction in error was concentrated on the players. At 25 Hz, the low-pass field improved players by 11.3% and the ball by 5.1%. At 6.25 Hz, it improved players by 2.2% and did not improve the ball ($+0.3\%$, $[-2.0, +2.6]$). We varied the width of the filter from 0 to 1 frames at 6.25 Hz in Appendix F to isolate its impact.

4.4 Per-agent Forward Processes

Giving the ball its own tailored noise field at 25 Hz made its accelerations more realistic than a shared noise field for the ball and the players, or using white noise for the ball. Table 6 shows the generated ball's distributional acceleration metrics compared to real tracking data at 25 Hz. Across three matched seeds, the agent-type-specific ball's median acceleration was 3.2 to 5.6 times closer to ground-truth than either the shared low-pass field or a ball left with white noise. Its W_1 distance was 1.31–1.47 against 1.88–2.09 for every alternative.

Among the models that trained appropriately, the agent-type-specific field was also the burstiest (defined in Section 3.3) at 6.6–8.2, against 2.7 for the shared low-pass field and 2.6 for a ball left with white noise. This is compared to the ground truth value of 31.0. The shared acceleration-level

Table 4: The same data at two sampling rates, for a low-pass field against white noise, metres, lower is better. The low-pass width is $\sigma = 0.5$ frames at both rates. Absolutes are not comparable across rates. Intervals that exclude zero are marked *.

rate	metric	white	low-pass	change [95% CI]
25 Hz	minSADE	2.444	**2.211**	-9.51% $[-12.12, -6.74]^*$
	avgSADE	3.478	**3.113**	-10.48% $[-12.22, -8.66]^*$
	avgSFDE	7.557	**7.039**	-6.85% $[-8.99, -4.65]^*$
	avgPADE	3.157	**2.800**	-11.30% $[-13.05, -9.47]^*$
	avgBADE	10.516	**9.981**	-5.09% $[-7.66, -2.58]^*$
6.25 Hz	minSADE	2.134	**2.076**	-2.70% $[-5.09, -0.23]^*$
	avgSADE	3.139	**3.082**	-1.84% $[-3.27, -0.37]^*$
	avgSFDE	7.004	6.959	-0.65% $[-2.25, +0.97]$
	avgPADE	2.818	**2.757**	-2.19% $[-3.57, -0.81]^*$
	avgBADE	10.188	10.215	$+0.27\%$ $[-2.00, +2.56]$

field scored higher still at 10.1–10.9, but that model did not train appropriately (see Section 4.2), and its extreme accelerations became too large at 5.70–7.63 against the ground-truth's 0.438.

This realism came at a cost in displacement error. Against the shared low-pass field, the agent-type-specific ball increased average scene error by 3.1% $[+1.7, +4.6]$. Against a model with white noise on both agent types, it reduced the average scene error by 7.6% $[-9.3, -6.0]$. The ball's own displacement error was statistically unchanged in both comparisons.

The change from a shared low-pass field to the agent-type-specific ball also worsened both the ball's displacement error and its realism at 6.25 Hz. On the Bundesliga benchmark, moving the ball from the players' low-pass field to its own tailored field raised the ball's displacement error by 10.5% $[+8.0, +13.2]$ and the averaged scene error by 3.7% $[+2.3, +5.1]$ (Table 5). At 25 Hz, the ball's error was unchanged when switching to its own tailored field. At 6.25 Hz, the agent-type-specific model overshot the real ball's jitter by 6.5 times while lowering burstiness.

Table 5: Displacement error of the agent-type-specific ball against each control, at both sampling rates, as paired percentage change. Positive favours the shared control. Intervals that exclude zero are marked *.

rate	metric	vs. shared low-pass	vs. shared white
25 Hz	avgSADE	$+3.13\%$ $[+1.65, +4.64]^*$	-7.63% $[-9.33, -5.95]^*$
	avgPADE	$+3.29\%$ $[+1.87, +4.71]^*$	-8.59% $[-10.25, -6.91]^*$
	avgBADE	$+2.17\%$ $[-0.79, +5.24]$	-1.20% $[-4.25, +1.96]$
6.25 Hz	avgSADE	$+3.66\%$ $[+2.34, +5.05]^*$	$+3.09\%$ $[+1.81, +4.45]^*$
	avgPADE	$+2.51\%$ $[+1.29, +3.80]^*$	$+1.70\%$ $[+0.55, +2.93]^*$
	avgBADE	$+10.51\%$ $[+8.01, +13.16]^*$	$+11.54\%$ $[+8.98, +14.21]^*$

Table 6: Acceleration magnitude $|\Delta^2 x|$ of the generated ball (m/frame^2). Absolutes are not comparable between the two blocks.

model	median	q_{99}	burstiness	W_1
25 Hz, three seeds				
ground truth	0.0141	0.438	31.0	0
agent-type-specific	**0.141–0.203**	**1.16–1.49**	**6.6–8.2**	**1.31–1.47**
shared low-pass	0.559–0.785	1.53–2.09	2.7	1.88–2.03
shared acceleration level	0.558–0.759	5.70–7.63	10.1–10.9	1.91–2.04
white ball field	0.725–0.902	1.91–2.35	2.6	1.99–2.09
6.25 Hz, one seed				
ground truth	0.1131	2.845	25.1	0
agent-type-specific	0.740	4.011	5.4	0.807
shared low-pass	**0.230**	**2.229**	**9.7**	**0.332**

5 DISCUSSION

The forward process in diffusion models is an interesting target for optimisation. Our best model improved on benchmark scores across a range of sports. This used velocity-level noise which is smoother than white noise. We also tested a range of field types, including blue noise which is more jagged than white noise, and resulted in increased error.

Our investigation suggests that the noise-field should be matched to the frequency of frame-to-frame movements in the training data. The clearest evidence is the rate experiment: the same data down-sampled loses most of the benefit of the low-pass field (10.5% against 1.8%). This may also explain the results of the basketball dataset. It has a lower sampling rate than the two other sports making its frame-to-frame movements rougher; therefore, the smoother noise fields were mismatched.

Ordering the fields by low-frequency power also orders the results favourably with regards to displacement error. One exception exists, which is acceleration-level noise. This has the most low-frequency power and failed outright (producing displacement error of 38.8 m on a 105×68 m pitch). Its participation ratio is 1.0 and Figure 2 shows that it is essentially a random constant offset. This is antithetical to the idea of diffusion models: a noising process with no stochastic variety leaves nothing suitable for a model to reverse. Therefore, its failure in this context does not indicate a limit on the low-frequency ordering but rather the importance of reporting participation ratio alongside it. The ideal noise field is more structurally similar to the data being modelled, but must still be noise.

The velocity-level field only reduced displacement error when it was trained with plain squared loss. The effect was lost when whitened loss was used. This is because velocity-level noise in the forward process changes slowly, and the current step is directly influenced by previous steps. However, the standardising process within whitened loss passes the error through the filter L^{-1}, which removes the correlation. Whitened loss is derived from the variational bound, so it suits models trained to maximise likelihood, as in Lee et al. (2022). However, we found that whitening increased displacement error by 7.9% and removed the benefit of the velocity-level field. In the other direction, giving a white noise model the loss weighting of the velocity-level field changed nothing. The weighting alone, therefore, does not help. In our design, the correlated noise in the forward process helps the denoiser to learn to recover trajectories, but whitening the loss undoes this.

Agent-types move in fundamentally different ways. Players move slowly and smoothly, while a ball is impulsive. We shaped the noise fields to meet the requirements of specific agent-types. We reduced displacement error in the players by having smooth noise in their forward process, which innately is closer to their movements. We made the ball more realistic by using a forward process where we spiked its amplitude at a small number of random frames and accumulated the noise twice to mimic the ball's behaviour. This resulted in a ball that more closely matched the ground-truth distribution of accelerations, in both magnitude and burstiness, which post-hoc smoothing was unable to do (Appendix H). However, this caused a small increase in displacement error. This suggests a trade-off between displacement error and realism in this model, but allows for tailoring to specific use cases. That said, the agent-type-specific model, incorporating both custom fields still outperformed the baseline model, which used white noise exclusively.

6 CONCLUSION

The forward process in multi-agent trajectory diffusion models often uses white noise as a default, but our results show that it is worth tuning to a specific use case. Shaping the noise to match the temporal structure of trajectory data makes the denoising task simpler for the denoiser, making it more able to generate plausible motion. With this change, our design set a new state of the art on a trajectory generation benchmark on multiple sports. Two principles follow. 1) the corrupting noise should resemble patterns in the data, but must still be time-varying enough to corrupt it. 2) the corrupting noise should also be matched to how the data are sampled and agent-type-specific behaviours. Future work should consider applications to other similar domains such as pedestrian modelling and robotics, and should also consider varying the diffusion schedule by agent-type to adjust when the players and ball resolve in diffusion time, affecting how they condition on each other.

AI USE STATEMENT

In this work, we used generative AI tools for planning while writing, light rephrasing, and as a coding assistant for data handling, model development, and data visualisation. We have reviewed all AI-assisted work. For example, coding has been reviewed manually throughout the development process and figures have been checked for technical accuracy. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses commercially licensed player-tracking data and the public JointDiff benchmark datasets. No personal data beyond on-pitch positions is used, no human-subject study was run, and we see no direct potential for harm in generative models of sports trajectories beyond the general considerations that apply to synthetic data.

REPRODUCIBILITY STATEMENT

All benchmark numbers are produced with the evaluation code released by Capellera et al. (2026) which was run unmodified on their data. Section 3.3 defines every metric and states the aggregation behind each percentage. Appendix A gives the full implementation details so each model can be rebuilt from its description. Section 3.2 specifies every noise field by its construction and its parameters. The fields are fixed before training and contain no learned parameters. Code cannot be released because the project is industry-funded; Section 3.2 and Appendix A specify every component.

Training is deterministic at a fixed seed on the machine that produced the Bundesliga results, so the seed standard deviations reported carry no run-to-run component. We tested this by reproducing our work a week after the initial experiments on the same device at the same fixed seeds and results were identical.

The Bundesliga, NFL, and NBA data are the public benchmark release of Capellera et al. (2026). Our 25 Hz tracking data is commercially licensed and cannot be released so the results that depend on it cannot be reproduced from public data. All other results can.

REFERENCES

Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 961–971. IEEE, 6 2016. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.110. URL http://ieeexplore.ieee.org/document/7780479/.

Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie S. Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Marin Biloš, Kashif Rasul, Anderson Schneider, Yuriy Nevmyvaka, and Stephan Günnemann. Modeling temporal data as continuous functions with stochastic process diffusion. In *Proceedings of the 40th International Conference on Machine Learning*. JMLR.org, 2023.

Guillem Capellera, Antonio Rubio, Luis Ferraz, and Antonio Agudo. Unified uncertainty-aware diffusion for multi-agent trajectory modeling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Guillem Capellera, Luis Ferraz, Antonio Rubio, Alexandre Alahi, and Antonio Agudo. JointDiff: Bridging continuous and discrete in multi-agent trajectory generation. In *International Conference on Learning Representations (ICLR)*, 2026.

Pascal Chang, Jingwei Tang, Markus Gross, and Vinicius C. Azevedo. How I warped your noise: A temporally-correlated noise prior for diffusion models. In *International Conference on Learning Representations (ICLR)*, 2024.

Xiusi Chen, Wei-Yao Wang, Ziniu Hu, David Reynoso, Kun Jin, Mingyan Liu, P. Jeffrey Brantingham, and Wei Wang. PlayBest: Professional basketball player behavior synthesis via planning with diffusion. In *ACM International Conference on Information and Knowledge Management (CIKM)*, 2024.

Giannis Daras, Mauricio Delbracio, Hossein Talebi, Alexandros G. Dimakis, and Peyman Milanfar. Soft diffusion: Score matching with general corruptions. *Transactions on Machine Learning Research*, 2023.

Fabian Falck, Teodora Pandeva, Kiarash Zahirnia, Rachel Lawrence, Richard Turner, Edward Meeds, Javier Zazo, and Sushrut Karmalkar. A fourier space perspective on diffusion models. May 2025. URL https://www.microsoft.com/en-us/research/publication/a-fourier-space-perspective-on-diffusion-models/.

Yuxiang Fu, Qi Yan, Lele Wang, Ke Li, and Renjie Liao. MoFlow: One-step flow matching for human trajectory forecasting via implicit maximum likelihood estimation based distillation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Roger Girgis, Florian Golemo, Felipe Codevilla, Martin Weiss, Jim Aldon D'Souza, Samira Ebrahimi Kahou, Felix Heide, and Christopher Pal. Latent variable sequential set transformers for joint multi-agent motion prediction. In *International Conference on Learning Representations (ICLR)*, 2022.

Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Emiel Hoogeboom and Tim Salimans. Blurring diffusion models. In *International Conference on Learning Representations (ICLR)*, 2023.

Xingchang Huang, Corentin Salaün, Cristina Vasconcelos, Christian Theobalt, Cengiz Öztireli, and Gurprit Singh. Blue noise for diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*, 2024.

Hyunsung Kim, Han Jun Choi, Chang Jo Kim, Jinsung Yoon, and Sang Ki Ko. Ball Trajectory Inference from Multi-Agent Sports Contexts Using Set Transformer and Hierarchical Bi-LSTM. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 4296–4307. Association for Computing Machinery, 8 2023. ISBN 9798400701030. doi: 10.1145/3580305.3599779.

Sang-gil Lee, Heeseung Kim, Chaehun Shin, Xu Tan, Chang Liu, Qi Meng, Tao Qin, Wei Chen, Sungroh Yoon, and Tie-Yan Liu. PriorGrad: Improving conditional denoising diffusion models with data-dependent adaptive prior. In *International Conference on Learning Representations (ICLR)*, 2022.

Seongju Lee, Junseok Lee, Yeonguk Yu, Taeri Kim, and Kyoobin Lee. MART: MultiscAle relational transformer networks for multi-agent trajectory prediction. In *European Conference on Computer Vision (ECCV)*, 2024.

Weibo Mao, Chenxin Xu, Qi Zhu, Siheng Chen, and Yanfeng Wang. Leapfrog diffusion model for stochastic trajectory prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Severi Rissanen, Markus Heinonen, and Arno Solin. Generative modelling with inverse heat dissipation. In *International Conference on Learning Representations (ICLR)*, 2023.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2022.

Vikram Voleti, Christopher Pal, and Adam M. Oberman. Score-based denoising diffusion with non-isotropic Gaussian noise models. *arXiv preprint arXiv:2210.12254*, 2022. NeurIPS 2022 Workshop on Score-Based Methods.

Chenxin Xu, Maosen Li, Zhenyang Ni, Ya Zhang, and Siheng Chen. GroupNet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Raymond A. Yeh, Alexander G. Schwing, Jonathan Huang, and Kevin Murphy. Diverse generation for multi-agent sports games. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

A Implementation details

All models use a width of 256, eight attention heads and a feed-forward width of 1,024. The continuous chain has $S = 50$ steps and the discrete chain 10, with a quadratic noise schedule, $\beta_s = \left(\text{linspace}(\sqrt{\beta_0}, \sqrt{\beta_S}, S)\right)^2$ from $\beta_0 = 10^{-4}$ to $\beta_S = 0.5$. Training uses AdamW with weight decay 0.01, learning rate 10^{-3} after 1,000 warm-up steps, gradient clipping at 1.0, batch size 16, and an exponential moving average of the weights with decay 0.999 for sampling. The learning rate is halved every 40 epochs on Bundesliga and every 20 on NFL and NBA, following JointDiff's schedule for each dataset. Bundesliga models train for 200 epochs of 235 batches with 180° rotation augmentation. The 25 Hz models train for 120 epochs of 500 batches on windows of 160 frames, 40 observed and 120 generated, which is the same 4.8 s generated horizon as the benchmark.

The NFL and NBA models train for 100 epochs of 594 and 1,792 batches respectively, matching the configuration JointDiff reports for those two datasets, and are scored at a single seed on the full test splits of 2,624 and 12,500 scenes. NFL scenes are 50 frames at 10 Hz with 23 agents; NBA scenes are 30 frames at 5 Hz with 11 agents. Both observe 10 frames.

Every baseline row in this paper is quoted as published by Capellera et al. (2026) in the unguided setting. We did not re-run any baseline. JointDiff released its benchmark and evaluation code but not its model, so the JointDiff row is its authors' number and our reproduction is reported separately as the ε-prediction row of Table 1.

B Sample diversity

We use averaged scene error to reward models that produce samples that are close to the ground-truth consistently as we assume that this results in plausible sequences. We use best-of-20 error to reward models that are able to produce at least one sample in 20 similar to the ground-truth without punishing a diverse range of samples.

As we change the prior, a narrower prior could narrow the samples produced by a model. This would manifest with a lower averaged error as the model optimises to simply replicate the ground-truth without diversity.

This is particularly relevant for the velocity-level noise. The participation ratio of this noise drops from 35.0 in the low-pass control to 2.7. For context, the flat acceleration-level noise that resulted in a model that was unable to learn had a participation ratio of 1.0. Simply, velocity-level noise contains far fewer independent temporal patterns than many other options reported in this paper, so may result in less diverse outputs.

We measured the pairwise distance between samples on the generated frames of 128 test scenes at three seeds per model. The change from our low-pass control to velocity-level causes only a reduction from 2.46 ± 0.05 m to 2.35 ± 0.04 m (mean $\pm$ seed standard deviation). Against the same control, best-of-20 scene error improves by 7.4% and averaged scene error improves by 4.2%, which is the opposite of what a collapsed sampler would produce. This suggests that our results are not an artefact of flattened diversity.

C NUMBER OF SAMPLING STEPS

Every number in this paper uses 11 network evaluations per sample, which is every fifth step of the 50-step chain plus a final step to the clean sample. To verify that our results were not an artefact of this, we re-scored the seed-46 white noise control and velocity-level model at every second step and at every step, giving 25 and 49 evaluations.

The advantage grew with more steps. On average scene error the velocity-level model reduces error relative to the white noise control by 4.8% at 11 evaluations, 5.9% at 25 and 6.3% at 49. On best-of-20 error the same comparison gives 8.4%, 11.0% and 11.5%.

D THE TRAINING OBJECTIVE

For noise with covariance $\Sigma = LL^\top$, weighting the error by Σ^{-1} is plain squared error on the whitened residual $L^{-1}(\varepsilon - \varepsilon_\theta)$. We test both directions of this pairing.

Whitening the loss of the velocity-level field increases averaged scene error by 7.9%, which removes the whole of that field's benefit. Applying the same weighting to a white field, which colours the loss without colouring the noise, changes nothing (-0.1%). The decrease in error comes from correlated noise trained specifically under the plain loss.

E LOW FREQUENCY POWER CORRELATION

We measured how well low-frequency power orders the fields with Spearman's rank correlation between each field's low-frequency power and its avgSADE in Table 3. We used a rank correlation because the relationship is monotone but not linear. Error falls steeply from blue to white noise, then gradually, and flattens at the pink and velocity-level plateau. Over the six fields that trained, $\rho = -0.94$ (exact two-sided $p = 0.02$ over all 720 orderings of six fields). Pearson's correlation on the same values is $r = -0.75$. We excluded the acceleration-level field, which did not train (Section 4.2). Including it gives $\rho = -0.21$.

F FILTER WIDTH

Table 7: Filter width at 6.25 Hz on the downsampled data, against the white noise control at the same rate. Test split data paired over 128 windows. The lag-one autocorrelation is a property of the filter and is measured before training.

width	lag-one autocorrelation	avgSADE	paired change
white	-0.001	3.139	—
$\sigma = 0.25$ frames (40 ms)	-0.001	3.164	$+0.8\%$ $[-0.6, +2.1]$
$\sigma = 0.50$ frames (80 ms)	$+0.261$	**3.082**	$\mathbf{-1.8\%}$ $[-3.3, -0.4]$
$\sigma = 1.00$ frames (160 ms)	$+0.779$	3.218	$+2.5\%$ $[+0.9, +4.1]$

The attenuation shown in the sampling rate comparison is not a mis-set filter. The width is fixed at $\sigma = 0.5$ frames. This spans four times more wall-clock time at the lower rate, so we performed a sweep at 6.25 Hz (Table 7). Halving it to $\sigma = 0.25$ frames is indistinguishable from white noise ($+0.8\%$, $[-0.6, +2.1]$), and doubling it to $\sigma = 1.0$ frames is significantly worse than white noise ($+2.5\%$, $[+0.9, +4.1]$).

Matching the width in seconds would require $\sigma = 0.125$ frames which is a kernel narrower than one frame. This would induce a lag-one autocorrelation of -0.001. At that width, the field is white noise so the field cannot be expressed at all. The field's width must therefore be set in frames. The best width in frames also depends on the data. On the Bundesliga benchmark, also at 6.25 Hz, $\sigma = 0.95$ frames reduced error more than $\sigma = 0.5$ (Table 3).

This only used the weak low-pass field, which has a low-frequency power of 0.23. On the Bundesliga benchmark, also at 6.25 Hz, the velocity-level field carries a low-frequency power of 0.81 and reduced averaged scene error by 5.1% (Table 1).

G COMPONENTS OF THE BALL'S FIELD

The ball's field is built in two steps. First, white noise is given a two-level amplitude: it is multiplied by $k = 25$ on a random fraction $p = 0.056$ of frames. Second, that noise is accumulated twice with a damping time of $\tau = 12$ frames, which is what puts it at the acceleration level.

We trained the field twice more, once with each step removed, and scored all three at seed 46 on the results of Table 6.

Removing the two-level amplitude changes almost nothing. The ball's median acceleration sits $0.99\times$ as far from ground truth $[0.97, 1.01]$, and W_1 is 1.43 against 1.43. Its one measurable effect is at the extremes, where it brings the 99th percentile 12% $[7, 17]$ closer to ground truth.

Removing the accumulation instead loses most of the benefit. With the two-level amplitude alone, the median acceleration is $3.31\times$ further from ground truth $[3.19, 3.44]$, and W_1 rises to 1.96 against the 2.07 of a ball left on white noise.

H SMOOTHING AND BURSTINESS

Magnitude statistics can be improved without training. This can be demonstrated by smoothing. We applied a centred moving average to the generated ball with white noise. A five-frame average takes W_1 from 2.07 to 1.29, and a 31-frame average takes it to 0.58, which is better than any trained model at 25 Hz.

We additionally report *burstiness* to address this, defined as the ratio q_{99}/q_{50} representing the relationship between the 99th percentile and median of the ball's accelerations. The smoothed approach barely moved this metric, from 2.6 to 2.8 at every width against the real ball's 31.0.

I PUBLISHED BASELINES IN FULL

Tables 1 and 2 show JointDiff, its predecessor U2Diff, and any published method with a lower error than JointDiff on at least one metric. The full sets of rows are below, all as published by Capellera et al. (2026).

Table 8: All published Bundesliga rows, metres, lower is better.

method	minSADE	avgSADE	minSFDE	avgSFDE
GroupNet (Xu et al., 2022)	4.78	5.76	9.58	11.63
AutoBots (Girgis et al., 2022)	3.33	5.93	5.57	11.46
LED^{IID} (Mao et al., 2023)	3.89	4.58	8.06	9.74
MART (Lee et al., 2024)	2.50	4.16	5.06	9.00
MoFlow (Fu et al., 2025)	2.51	4.21	5.08	9.24
U2Diff (Capellera et al., 2025)	2.69	4.21	5.46	9.44
JointDiff (Capellera et al., 2026)	2.47	3.66	5.02	8.29

Table 9: All published NFL (yards) and NBA (metres) rows, lower is better. LED is reported on basketball only.

	American football (NFL)				basketball (NBA)			
	SADE		SFDE		SADE		SFDE	
method	min	avg	min	avg	min	avg	min	avg
GroupNet	4.42	5.33	10.01	12.18	2.12	2.84	3.72	5.15
AutoBots	3.02	4.82	6.33	10.68	1.75	2.73	2.73	4.71
LED^{IID}	3.48	4.12	7.95	9.63	1.77	2.30	3.25	4.45
LED	—	—	—	—	1.63	3.83	2.99	6.03
MART	2.55	4.26	5.99	10.31	1.52	2.46	2.77	4.78
MoFlow	2.33	4.02	5.51	9.98	1.52	2.42	2.73	4.64
U2Diff	2.59	3.74	5.97	9.02	1.48	2.12	2.68	4.14
JointDiff	2.36	3.40	5.53	8.40	1.39	2.01	2.53	3.95

J QUALITATIVE COMPARISON

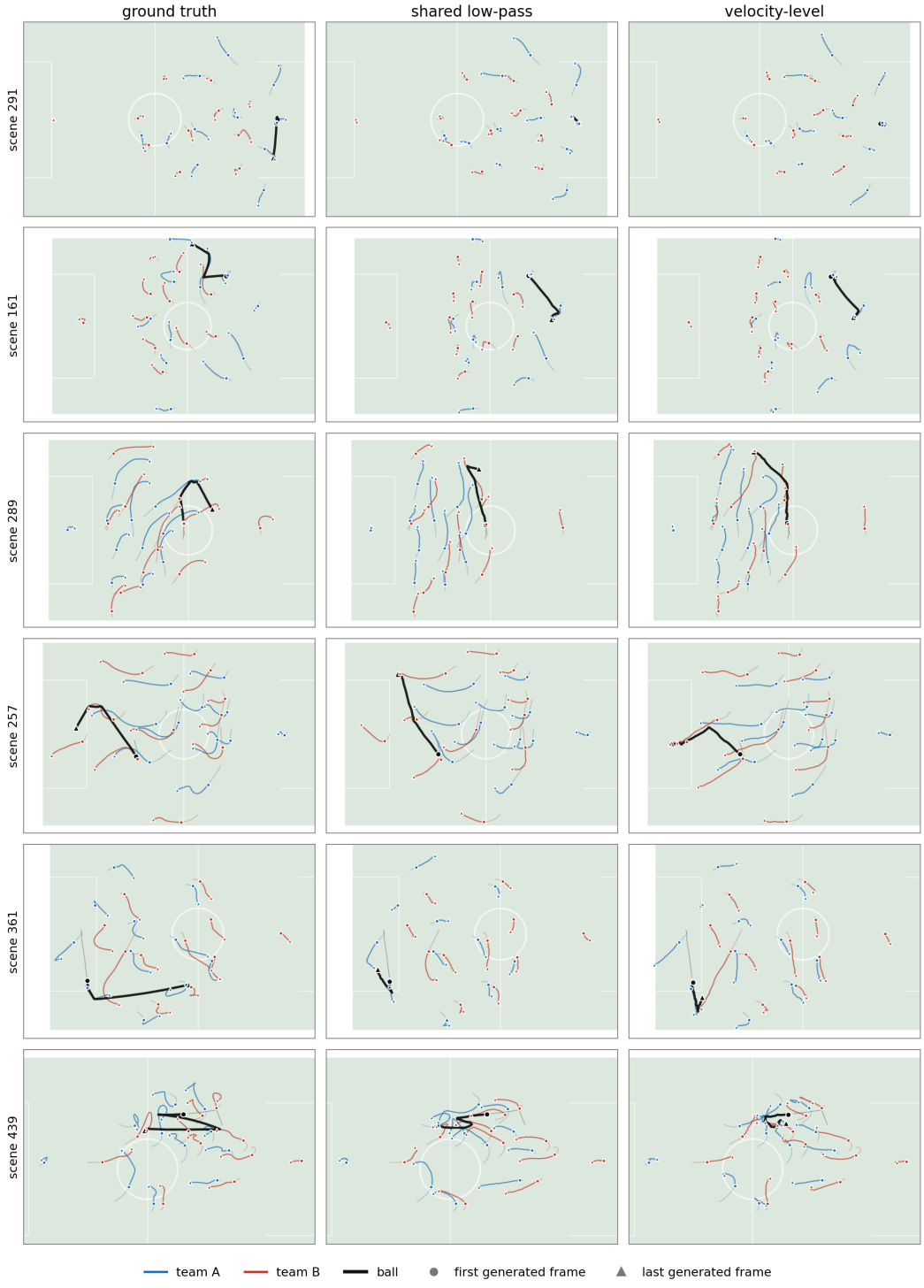

Figure 3: Six Bundesliga test scenes, chosen without reference to any model at evenly spaced quantiles of the ground-truth ball's path length over the generated frames: ground truth, the low-pass control, and our benchmark model of Table 1 (velocity-level noise field), both at seed 46. Each model panel shows a typical draw, the tenth best of twenty by scene error. Faint lines are the observed context and solid lines the generated horizon, with a circle at the first generated frame and a triangle at the last; the ball is drawn in black.