NARRATE TO SIMULATE: TRAJECTORY–LANGUAGE DIFFUSION FOR MULTI-AGENT SCENES

Anonymous authors
Paper under double-blind review

ABSTRACT

A simulated multi-agent scene is useful only if it can be read: not just *where* the agents go, but *what happens* between them. Generative models of multi-agent motion give the first and leave the second to the user. We extend joint continuous–discrete diffusion with a text modality: one denoiser generates the trajectories, the possession sequence and an event narration of a football play under one shared timestep, so every sample carries its own narration, grown from an empty sequence by an insertion-only edit flow. Insertion flows under-generate: a well-calibrated model reads its own short text as evidence of a short target and stops early, and more steps narrow the gap without closing it. LAG, a one-line training augmentation, repairs it: at ten calls it improves narration content by a fifth and CIDEr two- to threefold, surpassing the unaugmented model at fifty calls and matching it at a hundred. We evaluate fidelity to the real play, coherence between the streams and control through the narration. On a season of a top European league and public World Cup 2022 data, the co-generated narration costs no significant accuracy against a matched baseline without it, only ball tempo, which tracks narration length; each narration describes its own play about as faithfully as the reference describes the real one; and editing it steers the play: swapping a pass target for a nearby teammate moves the ball to that player in most samples, with no retraining or guidance. Upon acceptance we release the league data.

1 INTRODUCTION

A generated football play is only as useful as a person's ability to read it. Generative models of multi-agent trajectories have become remarkably good at answering *where*: given a few observed frames, socio-temporal forecasters (Yuan et al., 2021; Salzmann et al., 2020) and, more recently, diffusion and flow models (Mao et al., 2023; Jiang et al., 2023; Bae et al., 2024; Capellera et al., 2025; Xu & Fu, 2025; Fu et al., 2025) produce diverse, physically plausible continuations of a football or basketball play. But a coach, an analyst or a broadcaster does not reason in coordinates. They reason in sentences, such as *"#7 carries into the half-space, plays #9, who is dispossessed"*, and turning a tensor of trajectories back into such a sentence today takes a second step after simulation, by an analyst or a separate captioner that sees only the finished play. JointDiff (Capellera et al., 2026) took a first step toward interpretable simulation by attaching synchronous possession events to the trajectories and by letting text condition the motion, but the narration remained an input, never a joint output. A simulator that *writes while it plays* (Fig. 1) closes this gap: every sample carries the scene and its explanation, each constraining the other; the explanation is there for the thousand samples nobody watches; and, because it is text, it can be edited to steer what happens next.

To realize this we turn to discrete generative flows for text. The difficulty is not architectural, since a shared denoiser can host a text stream alongside trajectories and events, but one of representation and time: the motion is a structured socio-temporal tensor of fixed shape, agents by frames by coordinates, whereas a narration is a discrete sequence whose length is unknown in advance, and the two must be generated on the same denoising clock, so that a half-denoised scene is accompanied by a half-written narration the trajectories can attend to. Masked diffusion (Austin et al., 2021; Sahoo et al., 2024; Swerdlow et al., 2025) offers a fixed canvas of tokens but exposes the motion stream to a sea of placeholders for most of the process. Insertion-based Edit Flows (Havasi et al., 2025; Nguyen et al., 2025) instead grow the text from an empty sequence, so at every step the state is a partial narration with no placeholders: a natural fit for joint generation and, as we show, the formulation whose co-generated text couples best to the trajectories.

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.

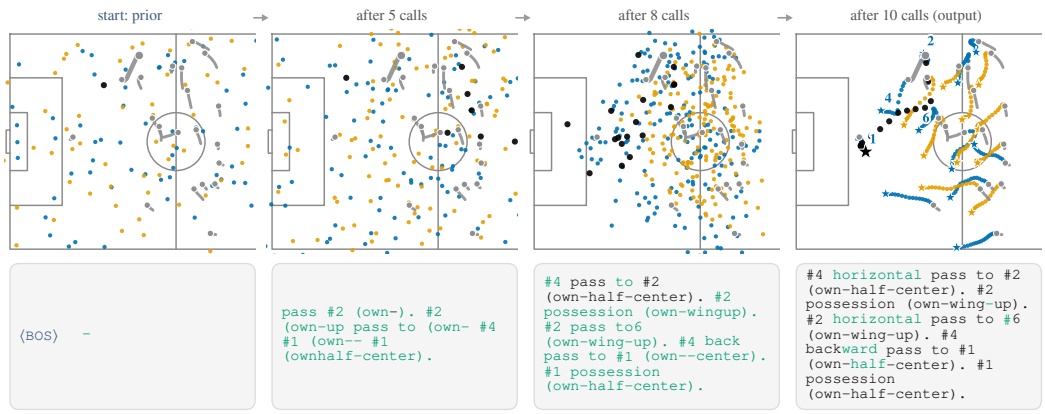


Figure 1: **One clock, two streams.** One denoiser completes a real 9.6-second League play and narrates it in the same ten calls, shown at the prior and after five, eight and ten calls. **Top:** the trajectories start from Gaussian noise and collapse into a play. Grey marks the ten observed frames, which are given; blue and orange mark the two teams and black the ball, drawn every third generated frame and ending in a star. The last panel labels the four players the narration names. **Bottom:** the narration starts empty (⟨BOS⟩) and grows by inserting tokens into the gaps of its current text; green marks the tokens inserted since the previous panel. Possession is also generated but not drawn; the supplementary video animates the process, possession included.

This choice uncovers a failure that is invisible in the continuous-time theory with exact rates and decisive in practice. An insertion sampler can at most double its length per call. At the ten-call budgets that make diffusion practical, the first steps therefore cannot keep pace with the schedule, and a well-calibrated model then reads its own short state as evidence of a *short target* and never catches up, even with more calls. We trace this collapse to two effects: a hard cap on how fast the text can grow, and a learned frozen ratio that keeps the early deficit until the end. The second is a train–inference mismatch of the same family as exposure bias (Bengio et al., 2015), and we repair it with LAG, which exposes the model during training to behind-schedule states; the change is one multiplication in the forward corruption, and the sampler is untouched.

We evaluate on a private season of a top European league (League) and on the public FIFA World Cup 2022 data along three axes. *Fidelity*: the co-generated narration comes at no significant cost in accuracy against the JointDiff baseline, with a measurable cost in ball tempo that follows the narration's length, and with ten calls LAG largely removes the collapse and beats the unaugmented model run with fifty on CIDEr and player mentions. *Coherence*: each narration describes its own play about as faithfully as the reference describes the real one. *Control*: editing the narration steers the play, and the length of the narration sets its tempo. In summary, our contributions are: **1)** a joint continuous–discrete–linguistic diffusion framework that co-generates trajectories, possession and event narration under a shared timestep, with insertion flows as the text modality; **2)** a diagnosis of length collapse in insertion flows (a doubling ceiling at small budgets and a frozen-ratio dynamic that persists at large ones) and LAG, a training-only augmentation that repairs the learned part, validated across seeds, budgets, datasets and against masked diffusion; **3)** an evaluation of the co-generated narration along fidelity, coherence and control against matched baselines, with a proper score that separates, per agent, genuine sharpening from mode collapse and a chain-based measure of whether a narration describes its own play; **4)** the League dataset and tooling to process the public World Cup 2022 corpus (a compact event grammar and a role-ordered agent layout), released upon acceptance.

2 Related work

Multi-agent trajectory generation. Generative modeling of team-sport motion has moved from recurrent and variational models (Zheng et al., 2016; Yeh et al., 2019; Li et al., 2021) through GANs and CVAEs (Gupta et al., 2018; Yuan et al., 2021; Xu et al., 2022) to transformer-based multi-modal predictors (Girgis et al., 2022; Alcorn & Nguyen, 2021; Lee et al., 2024) and, most recently, diffusion (Mao et al., 2023; Fu et al., 2025; Capellera et al., 2025). JointDiff (Capellera et al., 2026) added a synchronous discrete event stream and semantic control through possessor and

text guidance. We keep its continuous–discrete backbone and add a third, generated modality: the narration itself. These models are evaluated with best-of-k and average displacement errors (Alahi et al., 2016; Casas et al., 2020; Girgis et al., 2022), which respectively reward over- and under-dispersion; we add the energy score (Gneiting & Raftery, 2007), a proper scoring rule.

Discrete diffusion and flows for text. Absorbing-state diffusion (Austin et al., 2021) scaled to language modeling in MDLM and LLaDA (Sahoo et al., 2024; Nie et al., 2025), generating on a fixed token canvas by parallel unmasking. Edit Flows (Havasi et al., 2025) instead define a continuous-time Markov chain over insert, delete and substitute operations, growing text from an empty sequence with a Poisson rate per site. We adopt the insertion-only case and study its behavior when the continuous-time chain is discretized to a handful of steps, a regime these works do not study. The masked family has its own few-step remedies: remasking samplers (Wang et al., 2025a) let a committed token be masked again, GIDD (von Rütte et al., 2025) mixes uniform noise into the absorbing process so the model learns to correct its own tokens, and Zheng et al. (2025) trace part of the few-step degradation of masked models to imprecise categorical sampling. All three repair *substitution* errors on a canvas whose length is fixed up front; the insertion flow's few-step failure is in the *count*, and its sampler admits no corrector because committed tokens are permanent (our masked baseline runs without remasking). We are not aware of prior work that trains a discrete flow on deliberately behind-schedule states; the closest devices are the corrector terms of discrete flow matching (Campbell et al., 2024; Gat et al., 2024), which perturb the reverse chain rather than the training distribution, and, on the continuous side, input perturbation (Ning et al., 2023) and consistency regularization (Daras et al., 2023).

Train–inference mismatch. Autoregressive models suffer exposure bias (Ranzato et al., 2016), remedied by scheduled sampling, professor forcing and training on the learner's own state distribution (Bengio et al., 2015; Lamb et al., 2016; Ross et al., 2011); continuous diffusion loses fidelity at few steps, remedied by better solvers (Song et al., 2021) or by retraining for the sampler's states. LAG is the insertion-flow member of this family, with the distinction that we characterize the mismatch in closed form before repairing it.

Joint continuous–discrete generation. Mixed-modality diffusion has been explored for layouts (Levi et al., 2023) and CAD sketches (Chereddy & Femiani, 2025), and at scale for vision–language, where one network pairs image diffusion with text generated autoregressively (Zhou et al., 2025; Xie et al., 2025) or on a masked canvas (Li et al., 2025; Swerdlow et al., 2025; Yang et al., 2025b; You et al., 2026; Wang et al., 2025b), or composes per-modality diffusers (Tang et al., 2023). OneFlow (Nguyen et al., 2025) is the closest to ours: it generates text with an insertion-only edit flow concurrently with image flow matching. All of these operate in static domains and sample the text side with tens to hundreds of denoiser calls. Our setting is temporal and tri-modal: trajectories, possession and narration, generated jointly in a single pass, at a budget of ten calls shared with the motion, where masked models are known to degrade (Zheng et al., 2025; Wang et al., 2025a) and, as we show, insertion flows collapse in length. Our results add a structural argument, partial text against mask placeholders, for preferring insertion flows when a continuous stream attends to the discrete one.

3 METHOD

3.1 PROBLEM STATEMENT

We model a dynamic scene of N agents (ball and players) over T timesteps through three synchronous modalities:

- **Continuous.** Agent (x, y) locations $\mathbf{Y} \in \mathbb{R}^{T\times N\times 2}$.
- **Discrete–fixed.** Possession event denoting which player holds the ball at each timestep, $\mathbf{E} \in \{0, 1\}^{T\times N}$, one-hot per timestep, the ball's own index encoding "no possessor".
- **Discrete–variable.** A narration $\mathbf{W} = (w_1, \ldots, w_L)$ of L tokens from a vocabulary $\mathcal{V}$, describing the events of the play in order (e.g., `#7 pass to #9 (own-half-center), #9 pass failed, intercepted by #14`). Unlike $\mathbf{Y}$ and $\mathbf{E}$, its length L varies with the scene.

The scene is the tuple $\mathbf{X} = (\mathbf{Y}, \mathbf{E}, \mathbf{W})$. The conditional observations ("co") are $\mathbf{X}^{\text{co}} = (\mathbf{Y}^{\text{co}}, \mathbf{E}^{\text{co}}, \mathbf{W}^{\text{co}})$, where $\mathbf{Y}^{\text{co}} = \mathbf{M} \odot \mathbf{Y}$ and $\mathbf{E}^{\text{co}} = \mathbf{M} \odot \mathbf{E}$ are the entries selected by a binary mask $\mathbf{M} \in \{0,1\}^{T \times N}$ and the observed narration $\mathbf{W}^{\text{co}}$ is empty unless stated otherwise. The goal is to sample from $p(\mathbf{X} \mid \mathbf{X}^{\text{co}})$, i.e., to *complete the play and narrate it at once*. Observing the full trajectory gives captioning, and a prompt $\mathcal{G}$ text-guided generation (Capellera et al., 2026).

3.2 Joint Continuous–Discrete–Linguistic Diffusion

Following JointDiff, the forward process corrupts the modalities independently under a shared schedule of S steps: Gaussian diffusion on $\mathbf{Y}$ and multinomial diffusion on $\mathbf{E}$ (Ho et al., 2020; Hoogeboom et al., 2021). For the variable-length narration we use an insertion flow (§3.3) and tie its clock to the step index, $\bar{s} = 1 - s/S$ with $s \in \{0, \dots, S-1\}$ the noise level (s=0 clean), so that at every denoising step the three noisy states $(\mathbf{Y}_s, \mathbf{E}_s, \mathbf{W}_{\bar{s}})$ describe the same stage of the same play (Fig. 1). A single denoiser p_θ reads the joint noisy state and predicts the trajectory noise ϵ_θ, the event probabilities π_θ and, for each gap of the narration, the insertion quantities of §3.3.

Architecture. The denoiser is the JointDiff network with a text branch added; Appendix A gives its details, every loss term and all hyper-parameters. The observed $(\mathbf{Y}^{\text{co}}, \mathbf{E}^{\text{co}})$ and noisy $(\mathbf{Y}_s, \mathbf{E}_s)$ trajectories and possession form a $[T, N, \cdot]$ tensor processed by residual blocks, each a bidirectional state-space model along time per agent and set attention across agents per frame, from which two heads read ϵ_θ and π_θ. The text branch embeds the noisy narration $\mathbf{W}_{\bar{s}}$ with a T5-base encoder (Raffel et al., 2020) fine-tuned end to end and couples it to the motion through two cross-attentions. At the input, every agent's temporal tokens query the narration tokens and the result is added to the trajectory features, so the motion is denoised in view of the story so far; after the last block, the narration tokens query the $N \times T$ trajectory features and a linear head reads the per-site logits, rate and gate, so the next words are chosen in view of the denoised play. The JointDiff baseline is the same network without the text branch. Because insertion states contain only committed tokens, the trajectory attends to partial text without placeholders at every s (§4).

3.3 Insertion Flow for the Narration

We follow Edit Flows (Havasi et al., 2025) in the insertion-only form of OneFlow (Nguyen et al., 2025). The *forward* process reveals each token of $\mathbf{W}$ independently with probability $\kappa(\bar{s})$ and discards the rest, so the noisy state $\mathbf{W}_{\bar{s}}$ is an ordered subsequence of the target with expected length $\kappa(\bar{s})L$ (only `BOS` at $\bar{s}$=0, the full narration at $\bar{s}$=1). A *site* is a gap of the current text, before the first token, between two consecutive tokens or after the last, so a state of n tokens has n+1 sites. For every site i of $\mathbf{W}_{\bar{s}}$ the network predicts token logits, a Poisson insertion rate $\hat{\lambda}_i$, and a stop gate $\hat{p}_{0,i}$, the probability that nothing more is to be inserted at i. With k_i the number of target tokens still missing at site i and $\mathcal{T}_i$ those tokens, the insertion loss is:

$$\mathcal{L}^{\mathbf{W}} = \frac{\omega(\bar{s})}{L} \sum_i \Big[- \sum_{w \in \mathcal{T}_i} \log p_\theta(w \mid i) \;+\; \mathbf{1}[k_i{>}0] \big(\hat{\lambda}_i - k_i \log \hat{\lambda}_i \big) \;+\; \mathrm{BCE}\big(\hat{p}_{0,i},\, \mathbf{1}[k_i{=}0]\big) \Big], \quad (1)$$

whose three terms per site learn *which* words to insert (the negative log-likelihood of the missing tokens), *how many* (the Poisson negative log-likelihood of k_i under $\hat{\lambda}_i$, where something is missing) and *whether* to insert at all (a binary cross-entropy on the gate), weighted by the rate $\omega(\bar{s}) = \kappa'(\bar{s})/(1 - \kappa(\bar{s}))$ of the underlying continuous-time Markov chain (CTMC; Havasi et al., 2025). The *reverse* sampler inserts, per site and per step, at most one token, with probability:

$$p_i \;=\; (1 - \hat{p}_{0,i}) \cdot \min\big(1,\; \Delta\bar{s}\, \omega(\bar{s})\, \hat{\lambda}_i \big), \quad (2)$$

where $\Delta\bar{s}$ is the flow-time increment of one denoiser call ($\Delta\bar{s}$=$1/K$ for K calls), inserting the argmax token of the site's logits. Insertion-only implies no deletion and no substitution: committed tokens are permanent, and a state of n tokens (n+1 sites) can reach at most $2n$+1 tokens in one call.

Training objective and sampling. The joint loss is $\mathcal{L} = \lambda_{\mathbf{Y}} \mathcal{L}^{\mathbf{Y}} + \lambda_{\mathbf{E}} \mathcal{L}^{\mathbf{E}} + \lambda_{\mathbf{W}} \mathcal{L}^{\mathbf{W}}$, where $\mathcal{L}^{\mathbf{Y}} = \mathbb{E}\,\|\epsilon - \epsilon_\theta\|^2$ is the noise-prediction loss of DDPM on $\mathbf{Y}$, $\mathcal{L}^{\mathbf{E}}$ is the variational bound (a KL divergence between posteriors) of multinomial diffusion on $\mathbf{E}$, and $\mathcal{L}^{\mathbf{W}}$ is the insertion loss of Eq. 1 on $\mathbf{W}$. We use $\lambda_{\mathbf{Y}}$=$\lambda_{\mathbf{E}}$=1 and $\lambda_{\mathbf{W}}$=0.01, which brings the text term to the magnitude of the trajectory term. Sampling uses DDIM (Song et al., 2021) with skip ζ on $\mathbf{Y}$ and, through its multinomial posterior, on $\mathbf{E}$, and Eq. 2 on $\mathbf{W}$, all on the same $K = S/\zeta$ calls (Appendix A).

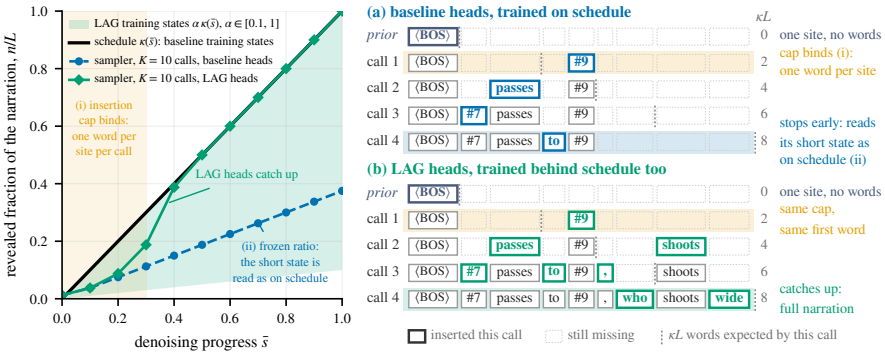


Figure 2: **Length collapse and LAG. Left:** Toy example with L=80, linear κ and K=10 calls. Baseline training only sees states on the schedule diagonal; LAG training also sees the shaded band below it, $\alpha\,\kappa(\bar{s})$ with $\alpha \in [\alpha_{\min}, 1]$, where a few-call sampler actually lives. Capped at one word per site per call (i), the baseline sampler falls below the schedule and then grows only by the schedule's ratio (ii, dashed); LAG heads catch up (solid). **Right:** The same mechanism on an eight-word narration and four calls. Both samplers place a single word at call 1, where the schedule asked for two. The baseline then grows by the schedule's ratio (1→2→3→4) and ends with the truncated "#7 passes to #9", while LAG uses every site (1→3→6→8) and delivers the whole sentence.

3.4 Why the Narration Collapses

At practical budgets, K=10 denoiser calls here, two effects compound, and the second does not vanish with more calls (Fig. 2-left). **(i) The cap starves the early steps.** Starting from BOS, the narration holds at most $\approx 2^k$ tokens after k calls, while a uniform schedule asks for L/K tokens per call: for L=80, K=10, the first call must place eight tokens at a single site. Eq. 2 saturates and the sampler leaves the schedule through no fault of the model. **(ii) Calibrated heads freeze the deficit.** Since unrevealed tokens are removed rather than masked, $\mathbf{W}_{\bar{s}}$ carries no trace of how much is missing; the main cue is the pair $(n, \bar{s})$, and in training these are locked together by $n \approx \kappa(\bar{s})L$. A calibrated rate head therefore infers $\hat{L} \approx n/\kappa(\bar{s})$ and predicts a *missing mass*, the expected number of tokens still to insert, $\sum_i (1 - \hat{p}_{0,i})\hat{\lambda}_i \approx n(1-\kappa)/\kappa$. Substituting into the expected step:

$$n(\bar{s}+\Delta\bar{s}) \;\approx\; n(\bar{s})\,\frac{\kappa(\bar{s}+\Delta\bar{s})}{\kappa(\bar{s})}, \tag{3}$$

the sequence grows by the *schedule's ratio* whatever its current length or the budget: any deficit, from the cap, from the heads' early under-prediction (0.85–0.86 and 0.95–0.96 at the first two calls, Appendix D.5) or from sampling noise, is carried unchanged to $\bar{s}$=1 (Appendix E.5). Under-generation is self-fulfilling precisely because the heads are right: teacher-forced on schedule, they predict 0.98–1.05 times the true missing mass from the third call on (Appendix D.5).

3.5 LAG: Behind-Schedule Augmentation

Effect (ii) is a mismatch between training and sampling: the sampler visits behind-schedule states that training never shows. We close it with one change to the forward corruption. With probability p_{LAG}, a training sample is revealed with probability $\alpha\,\kappa(\bar{s})$ instead of $\kappa(\bar{s})$, with $\alpha \sim \mathcal{U}(\alpha_{\min}, 1)$. Targets, weight $\omega(\bar{s})$ and sampler are unchanged; what changes is what the heads believe about the target length. From a state with n tokens at time $\bar{s}$, the baseline heads infer a single length n/κ; LAG heads have seen lengths up to $n/(\alpha_{\min}\kappa)$, so they predict more missing tokens, mostly on states that are behind schedule (Appendix D.5: at 0.3κ, 0.53–0.76 of the true missing mass without LAG, 0.82–1.01 with it; 1.03–1.08 on schedule). The per-site probabilities of Eq. 2 then approach one and the sampler uses its full doubling budget (Fig. 2, solid). LAG cannot lift effect (i), which bounds any model at very small budgets (Appendix E.5), nor catch up completely: a short state may mean "a short narration, on time" or "a long one, behind", and a calibrated model hedges between the two.

4 EXPERIMENTS

Datasets. Narration needs tracking and events on one clock and enough windows to train a text stream from scratch (Appendix B). **League** is one season of a top European league (228 matches, $208/10/10$ train/val/test) of 25 fps optical tracking with the provider's events, cut into 9.6 s windows (T=60 at 6.25 fps, N=23; $84{,}016/3{,}211/4{,}711$ windows) and narrated as compact event chains; prose captions and possessor sequences are also available. **World Cup** is PFF FC's public FIFA World Cup 2022 tracking with synchronized events, which we anchor to the tracking clock, render with the same grammar (actor, action, target, zone, pressure, stopper) and order by tactical role: 18,765 windows of 10.0 s over 52 matches ($41/5/6$), after removing twelve with misaligned or missing events. Its tracks are provider-smoothed, so every comparison is within one corpus.

Implementation. Unless stated otherwise, models observe the first ten frames and generate the other fifty, with S=50 steps, the quadratic noise schedule of Capellera et al. (2026) on $\mathbf{Y}$, cosine on $\mathbf{E}$ and linear κ. The DDIM skip ζ=5 gives $K=S/\zeta=10$ denoiser calls, "ten calls" below (five for ζ=10, fifty for ζ=1), nominal counts (network evaluations and parameter counts in Appendix A). The narration is embedded with a fine-tuned T5-base encoder (Raffel et al., 2020), and ours uses LAG with p_{LAG}=1, $\alpha_{\min}$=0.1 (sweep in §4.4). All train for 100 epochs, batch 32 on two A6000 GPUs (League) or 8 on one (World Cup). Each model reports one run, the same checkpoint at every budget: on League, the seed with the lowest validation loss where JointDiff and ours have several; World Cup five-seed means are in Appendix E.1.

Baselines. All share the windows and training regime (Appendix A). *JointDiff* is our network without the text stream, with the same loss weights and possession schedule; it does not narrate. *JointDiff → Ours (full play)* simulates, then narrates: our model trained with the whole play observed, the same text stream, narrates each of JointDiff's twenty plays in ten calls. *JointDiff + Guid*, trained separately, reads an event chain as a prompt through its cross-attention module CrossGuid (Capellera et al., 2026), with conditioning dropout, but does not narrate either; it is the prompted counterpart of ours in §4.3. *Masked diffusion* and the *insertion flow* are our joint model with the narration generated by absorbing-state diffusion on a fixed canvas, as in Dual Diffusion (Li et al., 2025) (Appendix A.1), or by the insertion-only edit flow of OneFlow (Nguyen et al., 2025) (§3.3) without LAG; with ours, they differ only in how the text is generated.

Metrics. We draw twenty samples per scene and ask three questions, one per table (definitions and a worked example in Appendix C). *Fidelity* (Table 1) scores each stream against the real play. For the trajectories, d is the scene-level average displacement error in metres (as is ES): sADE_{avg} for a typical sample, $\text{sADE}_{\min}$ for the best of 20; the energy score $\text{ES} = \mathbb{E}\, d(\hat{\mathbf{Y}}, \mathbf{Y}) - \frac{1}{2}\,\mathbb{E}\, d(\hat{\mathbf{Y}}, \hat{\mathbf{Y}}')$, with $\hat{\mathbf{Y}}, \hat{\mathbf{Y}}'$ two independent samples of a scene, is proper (Gneiting & Raftery, 2007), strictly so for each agent's position at each frame: it rewards a sharper sample set only when it is sharper around the truth. For possession, Acc_{gt} is the frame-wise accuracy. For the narration, CIDEr (Vedantam et al., 2015) and BERTScore (Zhang et al., 2020) score the wording against the reference; pF1 and eF1 score its content, over the players named (who) and the (actor, verb) events (who does what); and cF1_{gt} scores it against the real play, as the F1 between the narration's ball carriers (clause actors and pass receivers) and the play's possessors. *Coherence* (Table 2) scores one sample's streams against each other: Acc_{gen} checks possession against the possessor implied by the generated trajectories, cF1_{gen} the narration's carriers against the sample's own possession, and Order, the longest common subsequence of the two chains over the longer one, credits carriers named in sequence. *Control* (Table 3) measures how the play responds to the narration it reads: after an edit, how often and how closely the ball reaches the new target; for any narration, its length relative to the reference, the generated-to-real ball path ratio and ES.

4.1 FIDELITY TO THE REAL PLAY

Trajectories. Co-generating the narration with LAG costs no significant accuracy (Table 1): ours has the best typical sample on both corpora and the best energy score among the joint models, with best-of-20 and possession accuracy as good as JointDiff's or slightly below. Its tighter sample set is no gain under the energy score: over seeds JointDiff is slightly ahead on World Cup (2.72 ± 0.03 vs. 2.77 ± 0.05, p=0.09, though the run shown has the opposite sign) and on League (3.17 ± 0.07 vs. 3.25 ± 0.01; Appendix E.1). The measurable cost is in ball tempo, which §4.3 traces to narration length. The masked model, whose motion stream attends to a canvas of mostly placeholders until

Table 1: **Fidelity of each generated stream to the real play** (test, 20 samples). Lower is better for $\mathrm{sADE}_{\mathrm{avg}}$ (typical sample), $\mathrm{sADE}_{\mathrm{min}}$ (best of 20) and ES; higher for $\mathrm{Acc}_{\mathrm{gt}}$ and the narration scores against the reference chain (CIDEr, BERTScore, pF1, eF1) or the real play's carriers ($\mathrm{cF1}_{\mathrm{gt}}$; the reference chain scores 0.79 and 0.83). K: denoiser calls, nominally S/ζ (Appendix A); JointDiff → Ours narrates JointDiff's plays, whose cells are parenthesised and not ranked. **Bold** best, underline second within a corpus and budget (ten calls or 10+10; fifty); ours shaded. Absolute errors are not comparable across corpora; full table in Appendix E.1.

			trajectories (m)			possession	narration				
	Model	K	$\mathrm{sADE}_{\mathrm{avg}}$	$\mathrm{sADE}_{\mathrm{min}}$	ES	$\mathrm{Acc}_{\mathrm{gt}}$	CIDEr	BERTScore	pF1	eF1	$\mathrm{cF1}_{\mathrm{gt}}$
League	JointDiff (no narration)	10	5.72	**3.73**	**3.12**	0.359	–	–	–	–	–
		50	5.89	**3.79**	**3.09**	0.344	–	–	–	–	–
	JointDiff → Ours (full play)	10+10	(5.72)	(3.73)	(3.12)	(0.359)	**0.597**	**0.912**	**0.534**	0.45	0.46
	Masked diffusion	10	6.14	3.96	3.45	0.331	0.112	0.880	0.470	0.33	0.37
		50	6.53	4.04	3.49	0.318	0.167	0.897	0.457	0.37	0.36
	Insertion flow	10	5.73	3.96	3.46	0.350	0.213	0.897	0.455	0.38	0.42
		50	5.81	3.93	3.34	**0.350**	0.447	0.915	0.499	0.48	0.46
	Insertion flow + LAG (ours)	10	**5.59**	3.78	3.25	**0.362**	0.587	**0.912**	0.532	**0.47**	**0.47**
		50	**5.67**	3.89	3.31	0.349	**0.759**	**0.921**	**0.529**	**0.52**	**0.47**
World Cup	JointDiff (no narration)	10	4.94	**3.05**	2.76	**0.350**	–	–	–	–	–
		50	4.95	**3.05**	**2.61**	**0.335**	–	–	–	–	–
	JointDiff → Ours (full play)	10+10	(4.94)	(3.05)	(2.76)	(0.350)	0.706	0.924	0.506	0.47	0.48
	Masked diffusion	10	5.27	3.28	3.04	0.302	0.600	0.921	0.467	0.50	0.45
		50	5.27	3.21	2.85	0.283	0.740	0.930	0.448	0.50	0.43
	Insertion flow	10	4.82	3.15	2.82	0.327	0.375	0.911	0.435	0.44	0.45
		50	4.87	3.12	2.74	0.323	0.682	0.924	0.473	0.51	0.48
	Insertion flow + LAG (ours)	10	**4.71**	**3.05**	**2.71**	0.342	**0.785**	**0.925**	**0.526**	**0.51**	**0.51**
		50	**4.77**	3.09	2.70	0.330	**1.148**	**0.934**	**0.528**	**0.56**	**0.52**

late, hurts the trajectories on both corpora, while the plain insertion flow, which always offers partial text without placeholders, helps the typical sample on World Cup.

Narration. Among the joint models, ours writes the best narration on every content metric (pF1 0.53 on both corpora, against 0.44–0.47; eF1 0.47 and 0.51) and the highest CIDEr (0.59 and 0.79, against 0.11 and 0.60 for masked diffusion). Its content differs by corpus: League chains carry duels, tackles and dribbles at the reference rate, while World Cup chains are mostly the possession sequence with zones, with a presser in 17% of samples against 78% of the references (Appendix D.2). Narrating afterwards does not do better: our own text model, narrating each JointDiff play after ten extra calls to simulate it, writes a poorer narration on every metric on World Cup (CIDEr 0.71 vs. 0.79, pF1 0.51 vs. 0.53, eF1 0.47 vs. 0.51) and on League matches the wording (CIDEr 0.60 vs. 0.59, pF1 0.53) with fewer correct events (eF1 0.45 vs. 0.47). Co-generation thus buys a content-accurate story at no significant cost in accuracy (per metric and with the full play observed in Appendix D.1).

4.2 Coherence Between the Streams

A narration is only useful if it describes the play it comes with (Table 2). Ours agrees with its own possession ($\mathrm{cF1}_{\mathrm{gen}}$ 0.81 on League, 0.83 on World Cup) at the level of the reference chain against the real play (0.79, 0.83), although our narrations are shorter and name fewer carriers, and far above its agreement with the real play ($\mathrm{cF1}_{\mathrm{gt}}$ 0.47, 0.51 in Table 1): it describes its own sample rather than guessing the ground truth (precision and recall in Appendix E.2). A narration written after the play is simulated, by our full-play model with the whole play in view, agrees less: JointDiff → Ours reaches $\mathrm{cF1}_{\mathrm{gen}}$ 0.78 and Order 0.65 on both corpora, against our 0.81–0.83 and 0.68–0.72. Part of this gap, and most of the gap in recall (Table 13), reflects JointDiff's busier plays (4.2 and 3.9 successive ball carriers per play against our 3.7 and 3.4): narrating our own plays afterwards still falls 0.02 short in $\mathrm{cF1}_{\mathrm{gen}}$ and 0.01–0.04 in Order (Appendix E.2). The masked model, whose canvas is mostly placeholders, is less faithful to its play (0.54, 0.72). Likewise between trajectories and possession: both insertion models agree more than JointDiff ($\mathrm{Acc}_{\mathrm{gen}}$ 0.94–0.96 vs. 0.92–0.93), while the masked model is below it on League (0.91) and level on World Cup (0.93).

4.3 Control Through the Narration

Because narration and play are kept consistent, the narration can steer the play. We test this by *pinning* a chain to the text stream, that is, giving it at every denoising step instead of generating it, while trajectories and possession are sampled freely.

Table 2: **Coherence between the generated streams** (ten calls, 20 samples, runs of Table 1). Narration vs. poss.: cF1_{gen} and order (longest common subsequence) of the carriers a narration names against its own generated possession; traj. vs. poss.: Acc_{gen} of the generated possession against the possessor implied by the generated motion. JointDiff → Ours: JointDiff's plays narrated afterwards by our full-play model (its Acc_{gen} is JointDiff's, parenthesised). Italic: the reference chain against the real play, the ceiling (Acc_{gen}=1 by construction). **Bold** best, underline second among the models; ours shaded.

	League			*World Cup*		
	narration vs. poss.		traj. vs. poss.	narration vs. poss.		traj. vs. poss.
Model	cF1_{gen}	Order	Acc_{gen}	cF1_{gen}	Order	Acc_{gen}
JointDiff (no narration)	–	–	0.927	–	–	0.922
JointDiff → Ours (full play)	0.78	0.65	(0.927)	0.78	0.65	(0.922)
Masked diffusion	0.54	0.41	0.906	0.72	0.61	0.934
Insertion flow	0.73	0.63	**0.952**	0.79	0.69	**0.960**
Insertion flow + LAG (ours)	**0.81**	**0.68**	0.939	**0.83**	**0.72**	0.939
Reference vs. real play (ceiling)	0.79	0.67	1.000	0.83	0.73	1.000

Editing the story edits the play. We take the reference chain of a test window and swap the receiver of its last completed pass, the *old* target, with the *new* target, the teammate nearest to it at the last observed frame, throughout the chain (Fig. 3). With the edited chain pinned, the new target receives the ball in 91% of samples on League and 69% on World Cup, and is the first receiver, the edit itself, in 61% and 39%, against at most 29% and 10% (receiving, first) without the edit; the ball passes within 1.1 and 3.3 m of it instead of 8–10 m (Table 3), though the old target still receives in about a third of the samples (Appendix E.3). JointDiff + Guid, trained for exactly this, follows the edit more closely (96% and 77%; first 65% and 52%). Ours is never trained with a given narration: the chain is pinned only at test time, in a state training never shows (the full chain at every noise level), so its control is training-free, comes from the model that writes the narration, and falls within five to eight points of the dedicated model (four to thirteen on the first receiver). A random teammate, often far from the play, is followed less (Appendix E.3), because the observed frames constrain who can receive; the play follows the story as far as physically possible, without retraining, guidance or sampler changes.

The length of the story sets the tempo of the play. Pinned to the true narration, our model reaches an energy score well below the baseline (2.31 vs. 3.12, 2.07 vs. 2.76) and restores the ball's travel (92% and 98% of the real distance; Table 3). The true narration is oracle information, so the clean signal is the realism recovery: the model has the capacity, and the remaining gap is a narration gap. Our own narrations reach two thirds of their reference length and the play is correspondingly calmer: the ball travels 86% and 91% of the real distance against 96% and 99% for JointDiff, with a path-length W_1 three to six times JointDiff's (7.6 vs. 2.3, 5.1 vs. 0.8; Appendix D.4). JointDiff + Guid, prompted with the same chain, improves about as much (3.12 → 2.21, 2.76 → 1.99; Appendix D.3): the play responds to what the narration says, not to how it enters the model. Across the insertion models the ball falls short in proportion to how short the narration is (Appendix D.4). This likely explains Table 1's proper-score gap, which follows the narration's *length*, not its quality: fifty calls raise CIDEr without lengthening the narration (66 → 61%, 67 → 67%; Appendix E.5) and the gap stays, while a generated possessor sequence, complete by construction, ties the baseline on World Cup, though not on League.

4.4 LAG ABLATION

LAG leaves the model, loss weights and sampler untouched, so the insertion flow of Table 1 is its exact ablation. At ten calls it raises pF1 by 17–21% (0.455 → 0.532 on League, 0.435 → 0.526 on World Cup) and eF1 by 16–24%, doubles to triples CIDEr (0.21 → 0.59; 0.38 → 0.79, with no seed overlap over five seeds) and improves the energy score (3.46 → 3.25, 2.82 → 2.71), at one to two points of Acc_{gen} (Table 2); without it the narration stops after one or two events and the ball path stays short (Appendix E.6). The fifty-call rows show the collapse of §3.4 directly: from fifty to ten calls the plain flow loses about half its CIDEr and ours a quarter to a third, so ours at ten calls beats the plain flow at fifty; with S=200 levels the plain flow saturates at a hundred calls (CIDEr 0.96 vs. 1.31; Appendix E.5). Scaling the insertion rate ×4 at inference instead saturates at CIDEr 0.46 and 37% of the reference length on World Cup, against 0.79 and 67% with LAG, and League alike (Appendix E.4). The strongest setting, p_{LAG}=1 with $\alpha_{\min}$=0.1, gives the best narration on both corpora and is ours; the full sweep and the five-call ceiling are in Appendices E.4 and E.5.

Table 3: **Control through the narration** (ten calls, 20 samples). Text stream: nothing, the model's own narration, or the reference or edited chain of Fig. 3, read as a prompt by JointDiff + Guid and pinned at test time only by ours (never trained with a given narration). new recv. / new first: share of samples in which the new target holds the ball / is the first receiver after the passer; dist.: minimum ball distance to it (m); over the windows whose edited pass is generated (3,100 League, 1,824 World Cup). Length, path and ES are undefined for an edited play (no ground truth).

		League						*World Cup*					
Model	Text stream	new recv.	new first	dist.	length	path	ES	new recv.	new first	dist.	length	path	ES
JointDiff	No narration	0.29	0.09	9.0	–	0.96	3.12	0.28	0.09	9.6	–	0.99	2.76
JointDiff + Guid	Reference prompt	0.19	0.01	8.2	–	0.97	2.21	0.19	0.01	8.8	–	0.95	1.99
	Edited prompt	0.96	0.65	0.4	–	–	–	0.77	0.52	2.3	–	–	–
Ours	Own narration	0.26	0.09	9.9	66%	0.86	3.25	0.27	0.10	10.1	67%	0.91	2.71
	Reference pinned	0.19	0.01	8.4	100%	0.92	2.31	0.22	0.03	8.6	100%	0.98	2.07
	Edited pinned	0.91	0.61	1.1	–	–	–	0.69	0.39	3.3	–	–	–

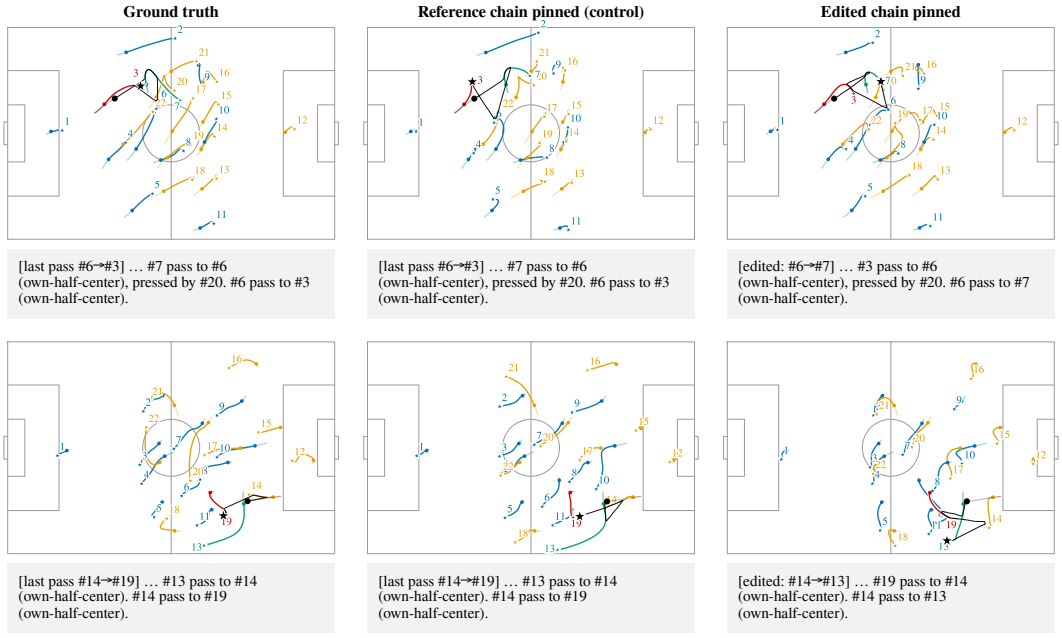


Figure 3: **Editing the story edits the play** (World Cup; last pass swapped with the nearest teammate). Left: ground truth and its chain; middle: reference chain pinned, the original target (red) is the first receiver; right: edited chain pinned, the new target (green) is; black: the ball. Text boxes show the end of the pinned chain, where the edited pass sits (best-case windows; Appendix E.3).

5 Conclusions

We extended joint continuous–discrete diffusion with a generated linguistic modality, so that one sample carries a play and its narration. Against a matched JointDiff baseline on two football corpora, the narration costs no significant accuracy (a proper score within noise on World Cup, ours slightly behind on League), only ball tempo, which traces to a learned coupling between the length of the story and the liveliness of the play: teacher-forcing the true story closes it. Each narration describes its own play at the faithfulness of the reference, better than one written after the play by the same text model, and editing a pass target sends the ball to the new player in up to 91% of the samples. Making the narration complete exposed a collapse of insertion flows, a combinatorial ceiling at small budgets compounded by a learned frozen-ratio dynamic that more steps do not undo; LAG, a one-line training augmentation, repairs it, improving narration content by a fifth and CIDEr two- to threefold at ten calls, past the unaugmented model's CIDEr at fifty. The framework applies wherever continuous dynamics and discrete descriptions co-evolve; Appendix F discusses limitations and next steps.

REFERENCES

Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 961–971, 2016.

Michael A Alcorn and Anh Nguyen. baller2vec: A multi-entity transformer for multi-agent spatiotemporal modeling. *arXiv preprint arXiv:2102.03291*, 2021.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993, 2021.

Inhwan Bae, Young-Jae Park, and Hae-Gon Jeon. SingularTrajectory: Universal trajectory predictor using diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17890–17901, 2024.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *International Conference on Machine Learning*, pp. 5453–5512. PMLR, 2024.

Guillem Capellera, Antonio Rubio, Luis Ferraz, and Antonio Agudo. Unified uncertainty-aware diffusion for multi-agent trajectory modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22476–22486, 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*, 2026.

Sergio Casas, Cole Gulino, Simon Suo, Katie Luo, Renjie Liao, and Raquel Urtasun. Implicit latent variable model for scene-consistent motion forecasting. In *European Conference on Computer Vision*, pp. 624–641. Springer, 2020.

Sathvik Reddy Chereddy and John Femiani. SketchDNN: Joint continuous-discrete diffusion for CAD sketch generation. In *International Conference on Machine Learning*, 2025. URL `https://openreview.net/forum?id=CxnKXLDCZM`.

Giannis Daras, Yuval Dagan, Alex Dimakis, and Constantinos Daskalakis. Consistent diffusion models: Mitigating sampling drift by learning to be consistent. In *Advances in Neural Information Processing Systems*, volume 36, pp. 42038–42063, 2023.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17282–17293, 2025.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *Advances in Neural Information Processing Systems*, volume 37, pp. 133345–133385, 2024.

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*, 2022.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024.

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*, pp. 2255–2264, 2018.

Marton Havasi, Brian Karrer, Itai Gat, and Ricky TQ Chen. Edit Flows: Variable length discrete flow matching with sequence-level edit operations. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Advances in Neural Information Processing Systems*, volume 34, pp. 12454–12465, 2021.

Chiyu Jiang, Andre Cornman, Cheolho Park, Benjamin Sapp, Yin Zhou, Dragomir Anguelov, et al. MotionDiffuser: Controllable multi-agent motion prediction using diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9644–9653, 2023.

Alex M Lamb, Anirudh Goyal, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set Transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pp. 3744–3753. PMLR, 2019.

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*, pp. 89–107. Springer, 2024.

Elad Levi, Eli Brosh, Mykola Mykhailych, and Meir Perez. DLT: Conditioned layout generation with joint discrete-continuous diffusion layout transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2106–2115, 2023.

Longyuan Li, Jian Yao, Li Wenliang, Tong He, Tianjun Xiao, Junchi Yan, David Wipf, and Zheng Zhang. GRIN: Generative relation and intention network for multi-agent trajectory prediction. In *Advances in Neural Information Processing Systems*, volume 34, pp. 27107–27118, 2021.

Zijie Li, Henry Li, Yichun Shi, Amir Barati Farimani, Yuval Kluger, Linjie Yang, and Peng Wang. Dual diffusion for unified image generation and understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2779–2790, 2025.

Weibo Mao, Chenxin Xu, Qi Zhu, Siheng Chen, and Yanfeng Wang. Leapfrog diffusion model for stochastic trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5517–5526, 2023.

John Nguyen, Marton Havasi, Tariq Berrada, Luke Zettlemoyer, and Ricky TQ Chen. OneFlow: Concurrent mixed-modal and interleaved generation with Edit Flows. *arXiv preprint arXiv:2510.03506*, 2025.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 50608–50646, 2025.

Mang Ning, Enver Sangineto, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Input perturbation reduces exposure bias in diffusion models. In *International Conference on Machine Learning*, pp. 26245–26265. PMLR, 2023.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations*, 2016.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Subham S Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 130136–130184, 2024.

Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *European Conference on Computer Vision*, pp. 683–700. Springer, 2020.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Alexander Swerdlow, Mihir Prabhudesai, Siddharth Gandhi, Deepak Pathak, and Katerina Fragkiadaki. Unified multimodal discrete diffusion. *arXiv preprint arXiv:2503.20853*, 2025.

Zineng Tang, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal. Any-to-any generation via composable diffusion. In *Advances in Neural Information Processing Systems*, volume 36, pp. 16083–16099, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4566–4575, 2015.

Dimitri von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion. In *International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=rvZv7sDPV9.

Guanghan Wang, Yair Schiff, Subham Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. In *Advances in Neural Information Processing Systems*, volume 38, pp. 147282–147339, 2025a.

Jin Wang, Yao Lai, Aoxue Li, Shifeng Zhang, Jiacheng Sun, Ning Kang, Chengyue Wu, Zhenguo Li, and Ping Luo. FUDOKI: Discrete flow-based unified understanding and generation via kinetic-optimal velocities. In *Advances in Neural Information Processing Systems*, volume 38, 2025b.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *International Conference on Learning Representations*, 2025.

Chenxin Xu, Maosen Li, Zhenyang Ni, Ya Zhang, and Siheng Chen. GroupNet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6498–6507, 2022.

Yi Xu and Yun Fu. Sports-Traj: A unified trajectory generation model for multi-agent movement in sports. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=9aTZf71uiD.

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

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. MMaDA: Multimodal large diffusion language models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 138867–138907, 2025b.

Raymond A Yeh, Alexander G Schwing, Jonathan Huang, and Kevin Murphy. Diverse generation for multi-agent sports games. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4610–4619, 2019.

Zebin You, Shen Nie, Xiaolu Zhang, Jun Zhou, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. LLaDA-V: Large language diffusion models with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9813–9823, 2021.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*, 2020.

Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. In *International Conference on Learning Representations*, 2025.

Stephan Zheng, Yisong Yue, and Jennifer Hobbs. Generating long-term trajectories using deep hierarchical networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. In *International Conference on Learning Representations*, 2025.

Appendix map. Each part points to the main text it supports.

- **A** Framework, architecture and implementation details of §3.
- **B** Datasets of §4, and what the event chains contain.
- **C** Metric definitions of §4, with one worked example.
- **D** Additional experiments behind claims of §3.4–§4.3.
- **E** Extended versions of Tables 1–3 and of §4.4.
- **F** Limitations and future work.

A Framework, Architecture and Implementation Details

Forward processes. The three modalities are corrupted independently on one clock of S levels $s \in \{0, \dots, S-1\}$ (larger is noisier), with flow time $\bar{s} = 1 - s/S \in \{1/S, \dots, 1\}$. Trajectories follow Gaussian diffusion (Ho et al., 2020): with a variance schedule $\{\beta_s\}$, $\alpha_s = 1-\beta_s$ and $\bar{\alpha}_s = \prod_{r \le s} \alpha_r$, $q(\mathbf{Y}_s \mid \mathbf{Y}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_s}\,\mathbf{Y}_0, (1-\bar{\alpha}_s)\mathbf{I})$, and the network predicts the added noise with $\mathcal{L}^{\mathbf{Y}} = \mathbb{E}\,\|\epsilon - \epsilon_\theta(\mathbf{X}_s, s)\|_2^2$, averaged over all valid entries (t, n), observed frames included. Possession follows multinomial diffusion toward the uniform distribution over the N agents (Hoogeboom et al., 2021), $q(\mathbf{E}_s \mid \mathbf{E}_0) = \mathrm{Cat}\big(\bar{\alpha}_s \mathbf{E}_0 + (1 - \bar{\alpha}_s)/N\big)$ under its own schedule; the network predicts $\hat{\mathbf{E}}_0 = \pi_\theta$, and $\mathcal{L}^{\mathbf{E}}$ is the KL divergence between $q(\mathbf{E}_{s-1} \mid \mathbf{E}_s, \mathbf{E}_0)$ and $q(\mathbf{E}_{s-1} \mid \mathbf{E}_s, \hat{\mathbf{E}}_0)$, replaced by the negative log-likelihood of $\mathbf{E}_0$ at $s{=}0$ (Capellera et al., 2026). The narration follows the insertion flow of §3.3: every token of $\mathbf{W}$ is kept independently with probability $\kappa(\bar{s})$ ($\alpha\,\kappa(\bar{s})$ for a LAG sample, whose state *lags* behind the schedule), `BOS` is always kept, the surviving tokens are packed to the left, and the loss weight is $\omega(\bar{s}) = \min\big(20,\ \kappa'(\bar{s})/(1 - \kappa(\bar{s}))\big)$. For LAG, with probability p_{LAG} a training sequence draws $\alpha \sim \mathcal{U}(\alpha_{\min}, 1)$, independently per sequence and iteration; otherwise $\alpha = 1$. Levels are drawn uniformly until every level has more than ten recorded losses, then with probability $p(s) \propto \big(\overline{\mathcal{L}_s^2}\big)^{1/2}$, where $\overline{\mathcal{L}_s^2}$ is an exponential moving average (weight 0.1) of the squared joint loss at level s (Nichol & Dhariwal, 2021); each sample's loss is divided by $p(s)$.

Denoiser. The denoiser is the JointDiff network (Capellera et al., 2026), built on the Social-Temporal blocks of U2Diff (Capellera et al., 2025), over $T{=}60$ frames and $N{=}23$ agent slots (the ball, then eleven players per team, ordered by role). *Input.* For every (t, n) the vector $[\mathbf{Y}^{\text{co}},\ \mathbf{E}^{\text{co}},\ \mathbf{Y}_s,\ \mathbf{E}_s,\ \mathbf{M}] \in \mathbb{R}^7$ is projected linearly to d, concatenated with a linear projection of a learnable slot embedding $\mathbf{a}_n \in \mathbb{R}^{64}$, and mapped from $2d$ back to d with a ReLU, giving $\mathbf{H} \in \mathbb{R}^{T \times N \times d}$. *Blocks.* Each of the two residual denoising blocks (RDB) of JointDiff computes:

$$\mathbf{H} \leftarrow \mathbf{H} + \mathrm{SAB}\big(\mathrm{Mamba}(\mathbf{H} + \mathbf{W}_b\,\gamma(s))\big),$$

where $\gamma(s) \in \mathbb{R}^{128}$ is a sinusoidal embedding of s followed by two SiLU-activated linear layers and $\mathbf{W}_b$ a per-block projection to d, broadcast over (t, n); Mamba is a bidirectional Mamba-1 (Gu & Dao, 2024) (state 16, convolution 4, expansion 2; forward and time-reversed outputs summed, then LayerNorm) run along time for each agent; SAB is two post-norm Transformer encoder layers (Lee et al., 2019; Vaswani et al., 2017) (8 heads, feed-forward $4d$, ReLU, dropout 0.1) run across the N agents of each frame, without positional encoding over agents. *Heads.* A shared $1{\times}1$ convolution $d \to d$ with ReLU feeds a zero-initialized $1{\times}1$ convolution that reads $\epsilon_\theta \in \mathbb{R}^{T \times N \times 2}$ and a $1{\times}1$ convolution $d \to 1$ whose logits are softmaxed over the N agents of each frame into π_θ (the ball slot stands for "no possessor"). In the text-guided models the CrossGuid module of JointDiff sits between Mamba and SAB: a frozen T5-base encoder (Raffel et al., 2020) embeds the prompt $\mathcal{G}$, every agent's temporal tokens attend to it, and conditioning dropout (0.25) trains the unconditional branch (guidance scale 1).

Text stream (Fig. 4). Narrations are tokenized with the T5-base SentencePiece tokenizer ($|\mathcal{V}| = 32{,}100$), which appends `EOS`; we prepend `BOS` (the sentinel `<extra_id_1>`), truncate to $L_{\max}$ tokens and right-pad. The noisy narration $\mathbf{W}_{\bar{s}}$ (the packed committed tokens; for the masked baseline, the canvas) is embedded by a second T5-base encoder, fine-tuned end to end, with padding masked in its self-attention, and projected linearly to $\mathbf{U} \in \mathbb{R}^{L \times d}$ (Fig. 4). Let $\mathrm{PE}_j \in \mathbb{R}^d$ be the sinusoidal encoding of index j (Vaswani et al., 2017) (one table for frame and token indices, dropout 0.1), $\mathbf{e}_n \in \mathbb{R}^d$ a second learnable slot embedding shared by both attentions, and $\mathrm{MHA}(\mathbf{Q}, \mathbf{K})$ an 8-head

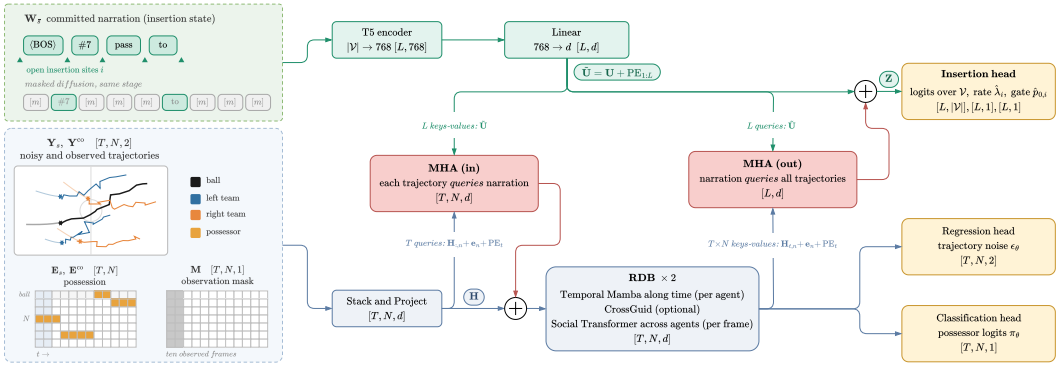


Figure 4: **The denoiser.** The JointDiff trunk (blue): the noisy and observed trajectories $\mathbf{Y}_s, \mathbf{Y}^{\text{co}}$, the possession $\mathbf{E}_s, \mathbf{E}^{\text{co}}$ and the observation mask $\mathbf{M}$ are stacked and projected to $\mathbf{H} \in \mathbb{R}^{T\times N\times d}$ and pass through two residual denoising blocks (RDB), each a temporal Mamba along time per agent, then, in the guided models only, CrossGuid, then a social Transformer across agents per frame. The narration branch (green): the committed narration $\mathbf{W}_{\bar{s}}$ with its open insertion sites i (for the masked baseline, the canvas with $[m]$ tokens at the same stage) is embedded by a T5 encoder and projected to $\mathbf{U} \in \mathbb{R}^{L\times d}$. Two cross-attentions couple the streams (red): at the input each agent's T tokens query the L narration tokens and the result is added to the trajectory features; at the output the L narration tokens query all $T\times N$ trajectory tokens and, with a residual from $\tilde{\mathbf{U}}$, give $\mathbf{Z}$, which feeds the insertion head; the arrows state what each attention reads, with positional encodings and slot embeddings added to its inputs only. The insertion head reads per-site token logits, insertion rate $\hat{\lambda}_i$ and stop gate $\hat{p}_{0,i}$. The regression and classification heads (yellow) read the trajectory noise ϵ_θ and the possessor logits π_θ. The JointDiff baseline is the same network without the green and red parts.

attention whose values equal its keys. At the input, before the first block, the T tokens of each agent n query the narration:

$$\mathbf{H}_{:,n} \leftarrow \mathbf{H}_{:,n} + \text{MHA}_{\text{in}}\big(\mathbf{H}_{:,n} + \mathbf{e}_n + \text{PE}_{1:T},\ \tilde{\mathbf{U}}\big) \in \mathbb{R}^{T\times d}, \qquad n = 1, \dots, N, \qquad \tilde{\mathbf{U}} = \mathbf{U} + \text{PE}_{1:L},$$

with padded narration positions excluded from the keys. At the output, after the last block, the narration queries all TN trajectory tokens, with a residual from its own input $\tilde{\mathbf{U}}$:

$$\mathbf{Z} = \tilde{\mathbf{U}} + \text{MHA}_{\text{out}}\big(\tilde{\mathbf{U}},\ \{\mathbf{H}_{t,n} + \mathbf{e}_n + \text{PE}_t\}_{t\le T,\, n\le N}\big) \in \mathbb{R}^{L\times d},$$

with the attention output zeroed at padded positions. A linear map $d \to |\mathcal{V}|{+}2$ then reads from $\mathbf{Z}$, at every site i (the gap after token i), the token logits, the rate $\hat{\lambda}_i = \text{softplus}(\cdot)$ and the gate $\hat{p}_{0,i} = \text{sigmoid}(\cdot)$; the masked baseline uses no padding masks and a map $d \to |\mathcal{V}|$. The text loss is Eq. 1 (Havasi et al., 2025): for every site i with k_i target tokens to insert after it, the negative log-likelihood of those tokens, the Poisson term $\hat{\lambda}_i - k_i \log \hat{\lambda}_i$ on sites with $k_i > 0$, and a binary cross-entropy of the gate against $\mathbf{1}[k_i{=}0]$; the sum is multiplied by $\omega(\bar{s})$ and divided by the target length L (BOS and EOS included). Targets are read off the packed state, so LAG samples need no change to the loss.

Sampling. With skip ζ the denoiser is called at the levels $s = S{-}1, S{-}1{-}\zeta, \dots \ge 1$, each of which inserts tokens, and once more at $s{=}0$, which inserts nothing and only completes $\mathbf{Y}$ and $\mathbf{E}$. A budget is named by its nominal $K = S/\zeta$ calls: $K{=}10$ ($\zeta{=}5$) is ten inserting calls at $s = 49, 44, \dots, 4$ plus the final one, i.e. 11 network evaluations, and $K{=}50$ ($\zeta{=}1$) is 50 evaluations, the last of which inserts nothing. Sampling starts from $\mathbf{Y} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, a uniformly random $\mathbf{E}$ and $\mathbf{W} = (\text{BOS})$. Between consecutive levels $s > s'$, $\mathbf{Y}$ takes a deterministic DDIM step (Song et al., 2021) ($\eta{=}0$) when $s - s' > 1$ and the DDPM posterior step otherwise, so $\zeta{=}1$ is ancestral sampling; $\mathbf{E}$ samples $q(\mathbf{E}_{s'} \mid \mathbf{E}_s, \hat{\mathbf{E}}_0{=}\pi_\theta)$, with an arg-max at $s{=}0$. $\mathbf{W}$ applies Eq. 2 at every site with $\bar{s} = 1 - s/S$ and $\Delta\bar{s} = \min\big((s - s')/S,\ 1 - \bar{s}\big)$ and inserts the arg-max token when both Bernoulli draws fire. Generated narrations are cut to $L_{\max}$ tokens and decoded without special tokens. The masked baseline's sampler is in Appendix A.1.

Baselines. Every JointDiff shares the cosine possession schedule and $\lambda_{\mathbf{E}}{=}1$ of the joint models, so it differs from them in the text stream only; matching the schedule moves the League baseline

by more than seed noise, hence the matched configuration throughout. *JointDiff → Ours (full play)* (Tables 1 and 2) uses no new training: our model trained with all sixty frames observed (insertion flow + LAG, event chains; the full-play row of Table 7 on World Cup; on League its best-validation checkpoint) reads each of JointDiff's 20 generated plays, trajectories and generated possession, as its observation and writes one narration per play in ten calls, so the k-th narration is scored against the k-th play. JointDiff + Guid (Table 3) adds CrossGuid text guidance (Capellera et al., 2026), trained with the window's event chain as the prompt (T5 encoder, conditioning dropout 0.25, guidance scale 1; prompts capped at 402 tokens on League and 129 on World Cup, truncating 0.9% of the training chains), and generates trajectories and possession only. Ours + Guid (Table 10) is ours with the same module, trained separately: it reads a prompt and still writes its own event chain; for Table 10, both Guid models are trained on League with the role headline and possessor sequence as the prompt. The masked baseline (Appendix A.1) shares the trunk, text encoder, both cross-attentions, $\lambda_{\mathbf{W}}$ and optimization with the insertion models; only its text process, loss and sampler differ.

Training and compute. Table 4 lists every setting. The denoiser has 5.8M parameters; the text stream adds the T5-base encoder (109.6M), the text projection and the two cross-attentions (0.7M) and the vocabulary head (8.3M), for 124.4M in total; the same encoder architecture is shared with JointDiff + Guid, so beyond it the text stream adds 9M. Training took about 14.5 h on two A6000 GPUs for the League text models (7.4 h for JointDiff) and about 6 h on one for World Cup (`torch.compile`), with no warm-up, EMA or early stopping. The validation loss, the joint objective averaged uniformly over the S levels, is computed every ten epochs, and the checkpoint minimizing it is evaluated.

Table 4: **Hyper-parameters.** Shared by every model unless a column states otherwise. $L_{\max}$ counts T5 tokens including `BOS` and `EOS`.

		League World Cup
Denoiser	residual blocks / model dim d / heads	2 / 256 / 8
	set-attention layers per block	2 (feed-forward $4d$, dropout 0.1)
	window T / observed frames / agent slots N	60 / 10 / 23
	step / slot embedding	128 / 64 (one per slot)
	temporal layer	bidirectional Mamba-1 (state 16, conv. 4, expansion 2)
	cross-attentions	2, 8 heads; sinusoidal PE (dropout 0.1); slot embedding in $\mathbb{R}^d$
	text embedder / tokenizer	T5-base encoder, fine-tuned / T5 SentencePiece, $\lvert\mathcal{V}\rvert$=32,100
	narration budget $L_{\max}$ (event chains)	200 160
	prompt encoder (guided models only)	T5-base encoder, frozen; cond. dropout 0.25
Diffusion	steps S	50
	$\mathbf{Y}$ schedule	quadratic: $\sqrt{\beta_s}$ linear in $[\sqrt{10^{-4}}, \sqrt{0.5}]$
	$\mathbf{E}$ schedule	cosine (all models)
	$\mathbf{W}$ reveal schedule $\kappa(\bar{s})$	linear; $\omega(\bar{s})$ clamped at 20
	timestep sampling	importance (Nichol & Dhariwal, 2021)
Loss	$\lambda_{\mathbf{Y}}$ / $\lambda_{\mathbf{E}}$ / $\lambda_{\mathbf{W}}$	1 / 1 / 0.01
	LAG: p_{LAG} / $\alpha_{\min}$	1 / 0.1 (ours); sweep in Table 15
	possession threshold (heuristic $\mathbf{E}$)	1.5 m
Optimization	optimizer	AdamW (β=(0.9, 0.999), weight decay 0.01), lr 10^{-3}, ϵ=10^{-4}
	schedule	$\times 0.5$ at epochs 20, 40, 60, 80; 100 epochs
	batch size × GPUs	32×2 8×1
	validation	joint loss, every 10 epochs; best checkpoint kept
	precision / gradient clipping	bf16 autocast / norm 5
	seeds	one run per model; appendix tables state their seeds
Sampling	skip ζ	5, i.e. K=10 (11 evaluations); 1 and 10 in Appendix E.5
	samples per scene	20
	text token choice	arg-max; insertions Bernoulli (Eq. 2)

A.1 Masked-Diffusion Baseline

The masked baseline replaces the insertion flow on $\mathbf{W}$ by absorbing-state diffusion (Austin et al., 2021; Sahoo et al., 2024), the text formulation of Dual Diffusion (Li et al., 2025); everything else is the joint model of Appendix A. The narration is a fixed canvas of $L_{\max}$ positions holding the right-padded token sequence (`BOS`, text, `EOS`, padding). Padding is part of the target, so the length is set by where the model places `EOS` and padding. *Forward.* At level s each position is independently replaced by the mask token $[m]$ (`<extra_id_0>`) with probability $m_s = (1-\epsilon)\,u_s$, $u_s = (s+1)/S$, $\epsilon = 10^{-3}$, i.e. $m_s = 1 - e^{-\sigma(u_s)}$ with $\sigma(u) = -\log\big(1-(1-\epsilon)u\big)$; a position is thus visible with

probability close to $\kappa(\bar{s})$, one level behind the insertion flow. *Loss.* With the SUBS parameterisation (Sahoo et al., 2024) (mask logit set to $-\infty$, visible positions copied):

$$\mathcal{L}^{\mathbf{W}} = \frac{1}{u_s\, L_{\max}} \sum_{l\,:\,w_{s,l}=[m]} -\log p_\theta(w_{0,l} \mid \mathbf{W}_s),$$

the continuous-time ELBO weight $\sigma'(u)/(e^{\sigma(u)}-1) = 1/u$: the same $1/(1-\kappa)$ form as ω, but unclamped and averaged over the canvas. *Sampling.* From an all-mask canvas, at each network evaluation (the same levels as the insertion models) the model predicts the arg-max token at every masked position; the positions visible at the next level s' are the already visible ones plus the most confident new predictions, up to $\lfloor L_{\max}(1-m_{s'})\rfloor$ in total ($L_{\max}$ after the last evaluation), and the rest stay masked. Visible tokens are never re-masked, so about $L_{\max}\zeta/S$ tokens are committed per evaluation. Decoding drops special tokens.

B DATASETS

Two conditions select the corpora. The events must be aligned to the tracking clock frame by frame, since the narration is scored against the possessor of every generated frame, and the corpus must hold enough windows to train a 256-dimensional text stream from scratch. The public NBA tracking releases carry play-by-play events that are not aligned to the tracking clock, and the NFL releases are aligned but hold too few narrated windows; the two football corpora below satisfy both.

Both corpora are windows of $T{=}60$ frames with $N{=}23$ agents (ball at slot 0, the team defending the left goal at slots 1–11 with its goalkeeper at 1, the other team at 12–22 with its goalkeeper at 12), coordinates in metres on a $105{\times}68$ m pitch with the broadcast-image convention (y down). Possession $\mathbf{E}$ is derived on both corpora with the geometric possessor of Capellera et al. (2026): the player nearest the ball when within 1.5 m, otherwise the ball index ("no possessor"). Players are referenced in every text modality by slot, `#1`–`#22`, so every mention is checkable against the trajectory tensor and player-mention metrics are well-defined.

B.1 FIFA WORLD CUP 2022

The source is PFF FC's public release of the 2022 World Cup[1] (**World Cup** below): 64 matches of 30 fps tracking with per-agent visibility flags and a separate event file.

Events are keyed by (period, frame, event id), joined from the event file on the identifier the tracking stream carries at each possession event's start frame; the event file's own timestamps (median offset 25–27 frames) are not used, nor are initial touches and bookkeeping types.

Windows. A window is 300 raw frames (10 s) with a hop of 150 (50% overlap), decimated by 5 to $T{=}60$ at 6 fps. Every raw frame must carry $11{+}11$ players and a ball, with no substitution and at least one rendered event in the window; on the decimated view, the mean `ESTIMATED` outfield share must be $\leq$0.6, frames with an estimated agent within 5 m of the ball $\leq$30%, frames with a player above 10 m/s $\leq$10% and with the ball above 40 m/s $\leq$2%, with no duplicated track.

Exclusions and split. We exclude ten matches whose event coordinates disagree with the tracking (median jersey-matched offset 0.8–3.9 m against $\approx$0.65 m in clean matches), two with extra-time events but no extra-time tracking, and the desynchronized extra-time periods of two of them. The remaining 52 matches are split by match, never by window (41/5/6 train/val/test), yielding 18,765 windows (14,563/1,594/2,608).

Channels. Players and ball come from the provider's smoothed channels (the raw player lists are 15 Hz sample-and-hold). The smoothed ball is placed on the annotated carrier and interpolated between events (within 1 cm of the actor at 91–95% of event frames), hence physically clean but partly a copy of the event labels. 44.7% of in-play player-frames are flagged `ESTIMATED`.

Difficulty. Table 5 shows that players travel as far in both corpora but with 4.5$\times$ the acceleration and 10$\times$ the jerk on League, and that the JointDiff baselines improve over constant velocity by a similar factor (32% on League, 28% on World Cup): the lower World Cup errors reflect the smoother data, not the model.

[1] `https://www.blog.fc.pff.com/blog/enhanced-2022-world-cup-dataset`

Table 5: **Dataset difficulty** (test splits, ten observed and fifty predicted frames at each corpus's native rate, 6.25 and 6 fps). Ground-truth player kinematics (speed, acceleration and jerk as first, second and third differences of position) and $\mathrm{sADE}_{\mathrm{avg}}$ of two trivial predictors against the trained JointDiff baseline (the run of Table 1).

	League	World Cup
player speed (m/frame)	0.40	0.35
player acceleration (m/frame^2)	0.127	0.028
player jerk (m/frame^3)	0.173	0.017
player path length over the window (m)	19.7	17.1
displacement from last observed position (m)	7.6	7.4
frames repeating the previous position	5.5%	0.0%
$\mathrm{sADE}_{\mathrm{avg}}$ (m), hold last position	7.61	7.37
$\mathrm{sADE}_{\mathrm{avg}}$ (m), constant velocity	8.45	6.87
$\mathrm{sADE}_{\mathrm{avg}}$ (m), constant velocity, players only	7.59	5.82
$\mathrm{sADE}_{\mathrm{avg}}$ (m), JointDiff baseline	5.72	4.94

Role-ordered slots. The left-defending team is found from the goalkeeper's mean x and each team is ordered by position group, back to front and left to right (GK; LB, LWB, LCB, MCB, RCB, RWB, RB; DM, LM, CM, RM, AM; LW, CF, RW), ties broken by jersey: per-slot role purity 82.3% against 41.9% for jersey order.

Event grammar. Each event is one clause: actor slot, verb (pass, cross, shot, duel, clearance, possession, rebound), target of a completed pass, the ball's zone in the actor's frame (`own`/`opp` × `box`, `half-center`, `wing-up/down`, `corner-up/down`), optional "`pressed by #N`", and a named stopper or duel outcome. Clauses are joined in time order, e.g.,

```
#16 pass to #20 (own-half-center), pressed by #7.  #20
tackle vs #7 (own-half-center), #20 wins the ball.  #9 pass
failed (opp-wing-up), intercepted by #14.
```

$L_{\max}$=160 tokens (99.9th percentile 156, maximum 179; 0.05% of chains truncated). Only this modality exists for World Cup.

B.2 League

A private season of a top European league; 228 matches (208/10/10 train/val/test) with 25 fps optical tracking and the provider's event and tactical annotations. Windows are 240 raw frames with a hop of 120, decimated by 4 to T=60 at 6.25 fps (9.6 s); windows shorter than 240 raw frames are dropped, leaving 84,016/3,211/4,711 windows.

Caption pipeline. A rule-based captioner turns the slot-anonymized window's events, tactics, phases and zone transitions into a tagged dense caption, which an LLM (Qwen3-32B, Yang et al., 2025a, served locally) rewrites into prose under a prompt that forbids dropping or adding players, zones and outcomes. A validation pass flags violations; the prompt and validator are released with the data.

Narration modalities. Four renderings of one test window:

- **Event chain**: the clauses of the World Cup grammar, plus throw-ins, free kicks, interceptions and recoveries; $L_{\max}$=200 (99.9th percentile 198, maximum 402; ≈0.1% truncated). Target of the joint models (Tables 1, 16) and condition of the text-conditioned baseline (Table 9):

  ```
  #14 possession, pressed by #9.  #14 backward pass
  to #16 (opp-wing-up).  #16 horizontal pass to #15
  (own-half-center).  #15 possession (own-half-center).  #15
  forward pass failed (own-half-center).
  ```

- **Prose caption**: a one-to-two-sentence summary with at most three slot tokens; the "captions" modality of Table 11:

  ```
  Player #14 passes back to Player #16, who passes
  horizontally to Player #15; Player #15 loses a forward
  pass.
  ```

- **Role headline**: the same arc without slot identifiers; with the possessor sequence, the prompt of the text-guided League models of Table 10:

  ```
  A wing back passes back to a centre back, who switches to
  another centre back, who loses a forward pass.
  ```

- **Possessor sequence**: the condensed chain of ball carriers (Tables 10 and 11): `#14, #16, #15`.

For caption targets the token budget is set from the data at training start.

Release. The corpus is private during review; every League experiment has a World Cup counterpart on public data. Upon acceptance, subject to the provider's clearance for (i), we release (i) the anonymized windows (role-ordered slots, no names, jerseys, clubs, match ids or absolute clock) with possession; (ii) all narration modalities plus a refined paragraph per window, keyed by window id, so §B.3 is reproducible without (i); (iii) the captioning pipeline and event grammar. The World Cup tooling (window extraction, event grammar and role-ordered slots) is released upon acceptance without the provider condition.

B.3 What the Event Chains Contain

A natural objection to co-generating an event chain is that a chain of passes is largely implied by the possessor sequence the model already generates. Table 6 classifies every event of the test-split references by its head verb. Completed passes and deliveries are indeed the majority of World Cup events and two fifths of League events, but the chains carry three kinds of information that no possessor sequence expresses: a second, off-ball player (the presser on 41% of World Cup and 27% of League events, the interceptor or blocker on a further 5–9%); the *outcome* of a play (a tenth of World Cup passes and one in seven League passes fail, with the manner of failure named); and the non-pass events (duels, tackles, dribbles, interceptions, clearances, rebounds and shots), which are 8–10% of events on both corpora and occur in a fifth of World Cup windows (21%) and over a third of League windows (37%). On League the pass direction (forward, horizontal, backward) and the carrier's possession under pressure are also spelled out. The structured recalls of §C score exactly these components, and the possessor-sequence row of Table 11 compares directly against a joint model that generates the possessor sequence instead of the event chain.

Table 6: **Content of the event-chain references** (test splits). An event is one sentence of the chain; families are assigned from the head verb. Qualifier shares are the fraction of events that carry the qualifier.

	World Cup	League
Test windows	2,608	4,711
Events per window	3.6	6.0
Share of events by family		
Completed pass / cross / long ball / set-piece delivery	80.7%	42.0%
Failed pass / cross (with outcome)	9.4%	7.0%
Possession / carry under pressure	1.6%	39.9%
Duel, tackle, dribble, challenge, 50-50	4.7%	4.6%
Interception, clearance, rebound	3.1%	4.3%
Shot / finish	0.4%	1.0%
Other	0.1%	1.2%
Events carrying each qualifier (independent, not a partition)		
"pressed by #N" (defender named)	41.3%	27.4%
"intercepted / blocked / defended by #N"	9.0%	5.5%
pass direction (forward / horizontal / backward)	0.0%	46.3%
pitch zone in parentheses	100.0%	86.8%

C Metric Definitions

Best-of-k sADE rewards over-dispersion and average sADE under-dispersion; the energy score, minimized by the true predictive distribution (Gneiting & Raftery, 2007), adjudicates between them, and the Wasserstein distance of the ball path adds the ensemble view that per-scene scores cannot see, at the price of ignoring conditioning. We report all four, with the spread. The metrics are

grouped below by the stream they score; every scene score is averaged over the test split, and every per-sample score over the $K{=}20$ samples unless stated otherwise.

Notation. For a test scene, $\mathbf{Y}$ are the ground-truth trajectories, $\{\hat{\mathbf{Y}}^{(k)}\}_{k=1}^{K}$ the samples, e_t the ground-truth possessor at generated frame t, $\hat{e}_t^{(k)} = \arg\max_n \pi_{t,n}^{(k)}$ the possessor of sample k, and $\mathcal{P}$ the set of generated agent-timesteps (t, n), every agent over the last 50 frames in the main setting. Displacement errors, energy scores, spreads and distances are in metres in every table.

C.1 Trajectories

Displacement (sADE). The scene ADE is $d(\hat{\mathbf{Y}}, \mathbf{Y}) = |\mathcal{P}|^{-1} \sum_{(t,n)\in\mathcal{P}} \|\hat{\mathbf{Y}}_{t,n} - \mathbf{Y}_{t,n}\|_2$ in metres, over all 23 agents. $\mathrm{sADE}_{\mathrm{avg}} = K^{-1} \sum_k d(\hat{\mathbf{Y}}^{(k)}, \mathbf{Y})$ is the error of a typical sample, $\mathrm{sADE}_{\min} = \min_k d(\hat{\mathbf{Y}}^{(k)}, \mathbf{Y})$ the error of the best of the K (an oracle choice).

Energy score and spread. With d as above:

$$\mathrm{ES} = \frac{1}{K}\sum_k d(\hat{\mathbf{Y}}^{(k)}, \mathbf{Y}) \;-\; \frac{1}{2}\,\mathrm{spread}, \qquad \mathrm{spread} = \binom{K}{2}^{-1} \sum_{k<k'} d(\hat{\mathbf{Y}}^{(k)}, \hat{\mathbf{Y}}^{(k')}), \tag{4}$$

the sample estimate of $\mathbb{E}\, d(\hat{\mathbf{Y}}, \mathbf{Y}) - \frac{1}{2}\mathbb{E}\, d(\hat{\mathbf{Y}}, \hat{\mathbf{Y}}')$. Because d averages per-agent, per-frame Euclidean distances, both expectations depend only on the marginal distribution of each $\hat{\mathbf{Y}}_{t,n}$: ES is the average of the energy scores of the $|\mathcal{P}|$ two-dimensional marginals, each strictly proper (Gneiting & Raftery, 2007). It is therefore proper for the scene, strictly so for each agent's position at each frame, but blind to the dependence across agents and frames; it scores every agent's forecast, not their coordination. Its first term is $\mathrm{sADE}_{\mathrm{avg}}$, minimized by the conditional mean and hence by under-dispersed sample sets; the spread term is what a collapsed ensemble forfeits. Like sADE, ES and spread are in metres. Lower is better.

Distributional realism (W_1). For the ball path length over the generated frames, we accumulate a fixed-range 50-bin histogram over all generated samples of the split and over the ground truth, and report the Wasserstein-1 distance between the normalized histograms on the bin centres, in metres. The ball *path ratio* (Tables 3 and 11) is the mean generated over the mean real ball path length; 1 means the ball travels as far as in the real plays.

C.2 Possession

Accuracy against the ground truth ($\mathrm{Acc}_{\mathrm{gt}}$) is the fraction of generated frames with $\hat{e}_t^{(k)} = e_t$, averaged over samples. **Accuracy against the generated play ($\mathrm{Acc}_{\mathrm{gen}}$)** replaces e_t by the geometric possessor of the sample's own trajectories $\hat{\mathbf{Y}}^{(k)}$ (the nearest player to the generated ball within 1.5 m, else no possessor) and therefore measures whether the possession stream and the trajectory stream of the same sample agree; the real data scores 1 by construction because its labels are that geometric rule.

C.3 Narration

Each sample's narration is scored against the single reference of its scene, averaged over samples and scenes. The metrics fall into three groups: how the text reads, what it says, and whether it matches a play. The last two are illustrated on one reference and one generated chain:

> *reference:* `#7 pass to #9 (own-half-center). #9 possession, pressed by #4. #9 pass to #11. #11 shot saved.`
> *generated:* `#7 pass to #11. #11 pass to #9, pressed by #2. #9 shot off target.`

Text similarity. CIDEr (Vedantam et al., 2015) ($n \le 4$) is computed once per sample index over the whole split, on whitespace tokens. BERTScore (Zhang et al., 2020) is the F1 of the default English model without baseline rescaling (its floor is about 0.9 on this grammar). The *relative length* is the

ratio of mean generated to mean reference narration length. References are the tokenized targets truncated at the text budget (0.05% of World Cup and 0.1% of League chains are affected).

Player-mention F1 (pF1). The set of slot identifiers (`#N`) in the generated narration against the set in the reference: precision, recall and their F1; an empty set on either side scores 0. In the example the reference names $\{7, 9, 4, 11\}$ and the generation $\{7, 11, 9, 2\}$: three of four on each side, precision 0.75, recall 0.75, pF1 0.75. pF1 asks *who* is mentioned, not what they did.

Event-level F1 (eF1). Each clause is mapped to an (actor, verb family) pair, the actor being its leading slot and the families being pass (with long balls, throw-ins, free kicks, kick-offs), cross, shot, duel (with challenges and 50-50s), tackle, dribble, interception, clearance, rebound and possession. Precision, recall and F1 are set overlaps of the generated and reference pairs, averaged over samples and then over scenes with a non-empty reference set. The example reference holds $\{(7, \text{pass}), (9, \text{possession}), (9, \text{pass}), (11, \text{shot})\}$ and the generation $\{(7, \text{pass}), (11, \text{pass}), (9, \text{shot})\}$; only $(7, \text{pass})$ is shared, so precision $1/3$, recall $1/4$, eF1 0.29. The same players in the wrong roles, which pF1 rewards, eF1 penalizes. A second variant, *eF1+target* ("+target" in Table 7), appends the first slot named after the verb (pass target or duel opponent): $(7, \text{pass}, 9)$ against $(7, \text{pass}, 11)$ no longer match.

Structured recalls score what a possessor sequence cannot express (§B.3). The *presser* set is the players named after "pressed by" ($\{4\}$ in the reference, $\{2\}$ in the generation: recall 0); the *action* set is the (actor, family) pairs whose family is a duel, tackle, dribble, challenge, 50-50, interception, clearance, rebound or shot ($\{(11, \text{shot})\}$ against $\{(9, \text{shot})\}$: 0); the *outcome* set is the (actor, phrase) pairs for the grammar's outcome phrases (failed, intercepted, blocked, saved, goal, off target, wins the ball, ...; $(11, \text{saved})$ against $(9, \text{off target})$: 0). Each recall is the fraction of the reference set recovered, averaged over samples and then over scenes with a non-empty reference set.

Narration against a play: cF1_{gt}, cF1_{gen} and order. These compare a narration not with the reference text but with a *play*, so they can be computed against the real play or against the sample's own generated play. A narration is reduced to its ordered chain of ball carriers (the actor of each clause and the target of each completed pass, consecutive repeats collapsed, pressers and stoppers excluded) and a play to the run-length chain of its possession stream over the generated frames (no-possessor frames dropped). Precision is the fraction of named carriers who hold the ball in that play, recall the fraction of ball holders who are named, cF1 their F1, and the *order* score the longest common subsequence of the two chains divided by the longer one; samples whose narration or play names no carrier are skipped. In the example the generated narration's chain is $7 \to 11 \to 9$. Against the real play, whose possession runs $7, 7, 9, 9, 9, 11, 11$ and hence has chain $7 \to 9 \to 11$, all three carriers are right ($\text{cF1}_{\text{gt}}{=}1$) but the order is not (longest common subsequence $7 \to 9$ or $7 \to 11$, two of three: order 0.67). If the sample's own generated possession runs $7, 7, 11, 11, 9, 9$, its chain is $7 \to 11 \to 9$ and the same narration scores $\text{cF1}_{\text{gen}}{=}1$ with order 1: it describes the play it came with, not the real one. Table 2 reports the generated-play version; Table 1 the real-play version, cF1_{gt}. The reference chain scored against the real play gives the ceiling row of Table 2.

Steering protocol. For Table 3 the edit swaps two teammates' slots throughout the chain (every $\#\text{old}$ becomes $\#\text{new}$ and vice versa; in the main text the new slot is the teammate nearest to the old one at the last observed frame, in Appendix E.3 also a random teammate, drawn with a fixed seed). "Receives the ball" means the slot appears in the play's carrier chain; "first" means it is the carrier right after the passer (or the first carrier if the passer never holds the ball); the ball distance is the minimum over generated frames of the distance between the generated ball and the player.

D Additional Experiments

D.1 Narration Quality in Detail

Extends the narration columns of Table 1 (§4.1). Forecasting from ten frames (Table 7), ours has the highest CIDEr, pF1 and eF1 of the three models on both corpora; masked diffusion names pressers more often on League (0.21 against 0.07) but almost no actions or outcomes. With the full play observed (World Cup), LAG helps as much as when forecasting: it raises CIDEr over the plain insertion flow from 1.18 to 1.81, pF1 from 0.67 to 0.78 and presser recall from 0.01 to 0.23, and

Table 7: **Narration quality** (ten calls unless stated, 20 samples). Ten frames: the forecasting models of Table 1, same runs. Full play (World Cup): the whole play is observed and only the narration is generated. Player mentions (pF1), structured recalls and events (eF1 and eF1+target) as in Appendix C. Our rows are shaded.

					player mentions			structured recall			events	
	Model	Observed	CIDEr	BERTScore	Prec	Rec	pF1	presser	action	outcome	eF1	+target
League	Masked diffusion	ten frames	0.112	0.880	0.657	0.397	0.470	0.21	0.00	0.04	0.33	0.13
	Insertion flow	ten frames	0.213	0.897	0.822	0.336	0.455	0.03	0.05	0.07	0.38	0.18
	Insertion flow + LAG (ours)	ten frames	0.587	0.912	0.680	0.466	0.532	0.07	0.10	0.11	0.47	0.23
World Cup	Masked diffusion	ten frames	0.600	0.921	0.593	0.411	0.467	0.05	0.00	0.00	0.50	0.17
	Insertion flow	ten frames	0.375	0.911	0.713	0.331	0.435	0.00	0.00	0.00	0.44	0.14
	Insertion flow + LAG (ours)	ten frames	0.785	0.925	0.656	0.466	0.526	0.06	0.01	0.01	0.51	0.22
	Masked diffusion	full play	0.973	0.927	0.768	0.617	0.666	0.20	0.00	0.00	0.62	0.44
	Insertion flow	full play	1.179	0.933	0.928	0.553	0.674	0.01	0.01	0.00	0.66	0.43
	Insertion flow + LAG (ours)	full play	1.809	0.941	0.914	0.700	0.776	0.23	0.03	0.07	0.67	0.46
	50 calls	full play	2.721	0.952	0.921	0.701	0.781	0.22	0.03	0.07	0.75	0.55

ours beats masked diffusion on every metric; fifty calls raise its CIDEr further ($1.81 \rightarrow 2.72$) and eF1 from 0.67 to 0.75, with pF1 and the structured recalls unchanged.

D.2 What the Generated Chains Contain

Extends the narration columns of Table 1. Table 8 applies the analysis of Appendix B.3 to the generations. On League, our narrations carry duels, tackles and dribbles at the reference rate (24% of samples against 20%) and failed plays at two fifths of it (14% against 35%); on World Cup they are mostly the possessor chain with zones, at two thirds of the reference length. Pressers are the hardest modifier on both corpora (14% and 17% of samples, against 83% and 78% of the references). Masked diffusion names a presser more often on League but recovers only a fifth of the reference pressers, and produces essentially no actions or outcomes (recall below 0.05). The missing modifiers are mainly a forecasting difficulty, not a limit of the text stream: with the full play in view, it recovers 23% of the reference pressers and 7% of the outcomes on World Cup (Table 7), against 6% and 1% when forecasting, and fifty calls move neither. What the co-generated text conveys reliably is *who* and *where*; pressers, outcomes and non-pass events are the headroom that longer or repaired narrations would add.

Table 8: **What the generated chains contain** (test splits, ten calls, 20 samples; the runs of Table 1). Share of samples whose narration names a presser, a failed play, or a duel, tackle or dribble; mean length in words; and the structured recalls of Appendix C. The reference row describes the references with the same extractors. Our rows are shaded.

		samples naming				structured recall		
	Narration	presser	failed play	duel, tackle, dribble	words	presser	action	outcome
League	Reference chain	83%	35%	20%	36	–	–	–
	Masked diffusion	68%	9.7%	3.8%	15	0.210	0.004	0.044
	Insertion flow + LAG (ours)	14%	14%	24%	24	0.067	0.103	0.113
World Cup	Reference chain	78%	27%	15%	24	–	–	–
	Masked diffusion	20%	0.1%	0.0%	15	0.048	0.000	0.000
	Insertion flow + LAG (ours)	17%	2.5%	0.1%	16	0.059	0.005	0.009

D.3 The Narration as a Prompt or Pinned

Extends §4.1 and §4.3, and the JointDiff rows of Table 3. JointDiff's prompt rows in Table 9 are the JointDiff + Guid of Table 3 (Appendix A). Ours are the checkpoints of Table 1: without a given chain they write their own narration; otherwise the chain is pinned in the text stream, whole from the first step or, *on schedule*, re-noised at every step to its $\kappa(\bar{s})$-subsequence, so the denoiser sees the partial text it was trained on and the complete chain only at the last call.

The play follows what the narration says, whichever way it enters the model. Given the reference chain, both models improve sharply (energy score 2.21 and 2.31 on League, 1.99 and 2.07 on

Table 9: **The narration given as a prompt or pinned** (ten calls, 20 samples). Text stream: what the model reads while it generates the play: nothing, an empty prompt, its own narration, or the reference chain (an oracle: the chain describes the future). Every chain is an event chain. JointDiff + Guid reads a chain as a prompt (its no-narration row is the plain JointDiff); ours writes its own narration or has the chain pinned in its text stream, whole from the first step or, on schedule, revealed on the $\kappa(\bar{s})$ schedule it was trained on.

		League			*World Cup*		
Model	Text stream	sADE_{avg}	ES	Acc_{gt}	sADE_{avg}	ES	Acc_{gt}
JointDiff	No narration	5.72	3.12	0.359	4.94	2.76	0.350
JointDiff + Guid	Empty prompt	5.98	3.29	0.360	4.91	2.80	0.331
	Reference prompt	3.95	2.21	0.631	3.54	1.99	0.494
Ours	Own narration	5.59	3.25	0.362	4.71	2.71	0.342
	Reference pinned	4.06	2.31	0.607	3.84	2.07	0.459
	on schedule	4.22	2.32	0.549	4.02	2.12	0.422

World Cup, against 3.12 and 2.76 for JointDiff without narration): ours, never trained with a given narration, comes within 0.1 of JointDiff + Guid, and revealing the pinned chain on the training schedule gives the same result (2.32 and 2.12), so the small gap is not a distribution shift. The narration's content thus drives the play; what our model adds is that it writes the narration in the same pass and keeps its streams coherent (§4.2).

Controlled generation. Table 10 compares JointDiff + Guid with Ours + Guid, a separately trained copy of ours that reads the reference narration as a prompt instead of having it pinned, and still writes its own chain (Appendix A). The two models use the prompt about equally well (energy score 2.46 and 2.44 on League, 1.99 and 2.02 on World Cup).

Table 10: **Controlled generation** (ten calls, 20 samples). Text stream: the reference narration, read as a CrossGuid prompt (not pinned) by JointDiff + Guid and by Ours + Guid, a separately trained copy of ours that also generates its own event chain (Appendix A). Input, stated per corpus under its name: on League the prompt is the role headline with the possessor sequence, not the event chain of Table 9; on World Cup it is the event chain.

		League role headline + possessors			*World Cup* event chain		
Model	Text stream	sADE_{avg}	ES	Acc_{gt}	sADE_{avg}	ES	Acc_{gt}
JointDiff + Guid	Reference prompt	4.34	2.46	0.556	3.54	1.99	0.494
Ours + Guid	Reference prompt	4.31	2.44	0.562	3.63	2.02	0.496

D.4 Narration Length and Ball Motion

Extends the tempo paragraph of §4.3. Table 11 ties the ball's tempo to narration length: across the

Table 11: **Narration length against ball motion** (ten calls, 20 samples). Length: generated narration length in words relative to the model's own reference. Path: generated over real mean ball path length; W_1: its Wasserstein-1 distance to the real distribution, in metres; ES: energy score, in metres. One run per model, that of Table 1 where the model appears there. The last two rows are insertion models without LAG that narrate another target. Our row is shaded.

	League				*World Cup*			
Model	length	path	W_1	ES	length	path	W_1	ES
JointDiff (no narration)	–	0.96	2.34	3.12	–	0.99	0.83	2.76
Masked diffusion	41%	0.84	9.09	3.45	62%	0.89	5.85	3.04
Insertion flow	26%	0.65	18.50	3.46	30%	0.80	11.07	2.82
Insertion flow + LAG (ours)	66%	0.86	7.61	3.25	67%	0.91	5.10	2.71
Insertion flow, other narration target								
possessor sequence	87%	0.93	3.85	3.31	92%	0.93	4.09	2.73
prose captions	65%	0.91	4.85	3.24	–	–	–	–

insertion models, the shorter the narration, the shorter the ball path. Among the models that narrate event chains, ours has the most realistic ball path on both corpora (W_1 7.6 and 5.1, against 9.1 and 5.9 for masked diffusion and 18.5 and 11.1 for the plain insertion flow), and the possessor-sequence model, whose narration comes out near its reference length (87–92%), is closest to JointDiff. On World Cup it also ties JointDiff's energy score (2.73 vs. 2.76); on League it does not (3.31 vs. 3.12).

D.5 Calibration of the Rate Head

Extends the teacher-forced probe of §3.4. We teacher-force the checkpoints of Table 1 on noisy states drawn from the training forward process. For the first 512 test windows and each of the ten flow times visited by the ten-call sampler, $\bar{s} \in \{0.02, 0.12, \ldots, 0.92\}$, all three modalities are noised at level $s = S(1-\bar{s})$; each narration token is kept with probability $\kappa(\bar{s})$ (on schedule) or $0.3\,\kappa(\bar{s})$ (behind schedule), BOS always kept. The factor 0.3 is a fixed test condition, the same for every model and unrelated to the training range of LAG ($\alpha_{\min}$=0.1 for ours): it is about where the plain flow's own samples end (26 and 31% of the reference length, Table 15). One denoiser call gives the predicted missing mass $\hat{M} = \sum_i (1-\hat{p}_{0,i})\,\hat{\lambda}_i$ over the sites of the state, which we compare with the true number of missing tokens M through $\mathbb{E}[\hat{M}]/\mathbb{E}[M]$ over windows. Without LAG and on schedule, the ratio is 0.98–1.05 for $\bar{s} \geq 0.22$ on both corpora; the first two calls under-predict (0.86 and 0.96 on League, 0.85 and 0.95 on World Cup). Behind schedule it falls to 0.53–0.76. Ours (p_{LAG}=1, $\alpha_{\min}$=0.1) lifts the behind-schedule ratio to 0.82–1.01 at the cost of a slight over-prediction on schedule (1.03–1.08 for $\bar{s} \geq 0.22$). The heads are right on the states they were trained on, and LAG teaches them the states the sampler visits.

E Additional Results

E.1 Table 1 in Full

Extends Table 1 (§4.1). Table 12 gives Table 1 on World Cup with every metric, as means over five seeds per model. Over seeds, ours has the best narration on every text metric (CIDEr 0.79 ± 0.04, against 0.46 for masked diffusion and 0.40 for the plain insertion flow) and, among the models that narrate, the best energy score (2.77, against 2.86 and 3.04). Against JointDiff, the average sADE is tied (4.82 ± 0.10 vs. 4.91 ± 0.07, Welch p=0.13) and JointDiff's proper-score edge is not significant (2.72 ± 0.03 vs. 2.77 ± 0.05, p=0.09). On League, Table 1 shows the seed with the lowest validation loss, of two for JointDiff and three for ours; the three seeds of ours are in Table 15. Over these seeds, ours has the lower average sADE (5.65 ± 0.06 vs. 5.75 ± 0.05) and JointDiff the lower energy score (3.17 ± 0.07 vs. 3.25 ± 0.01).

Table 12: **Table 1 in full** (World Cup, ten calls, 20 samples; mean±std over five seeds per model). The metrics of Table 1 except cF1_{gt}, which is computed only for the run Table 1 shows, plus the spread $\mathbb{E}\,d(\hat{\mathbf{Y}}, \hat{\mathbf{Y}}')$, a diagnostic with no better direction, never marked, and Acc_{gen} (Table 2). **Bold** best, underline second best. Our row is shaded.

	trajectories (m)				possession		narration			
Model	sADE_{avg}	sADE_{min}	ES	spread	Acc_{gt}	Acc_{gen}	CIDEr	BERTScore	pF1	eF1
JointDiff (no narration)	4.91 ±0.07	**3.05** ±0.02	**2.72** ±0.03	4.40 ±0.09	**0.348** ±0.003	0.923 ±0.001	–	–	–	–
Masked diffusion	5.24 ±0.11	3.28 ±0.08	3.04 ±0.18	4.39 ±0.16	0.315 ±0.009	0.917 ±0.016	0.457 ±0.086	0.915 ±0.003	0.464 ±0.011	0.48 ±0.01
Insertion flow	**4.82** ±0.03	3.17 ±0.03	2.86 ±0.06	3.91 ±0.07	0.322 ±0.007	**0.961** ±0.002	0.402 ±0.020	0.913 ±0.001	0.446 ±0.007	0.44 ±0.01
Insertion flow + LAG (ours)	**4.82** ±0.10	3.09 ±0.05	2.77 ±0.05	4.11 ±0.10	0.341 ±0.006	0.939 ±0.002	**0.789** ±0.042	**0.926** ±0.001	**0.524** ±0.007	**0.51** ±0.01

E.2 Coherence in Full

Extends Table 2 (§4.2). Table 13 extends Table 2 with carrier precision and recall. Ours has the highest cF1 (0.81, 0.83) and the highest recall (0.87, 0.88) of the models: it names more of its own

play's carriers than any other, while the plain insertion flow is more precise on World Cup (0.86 against 0.82) but misses more of them (0.77).

Simulate, then narrate. Narrating a finished JointDiff play with our own full-play model is as precise as co-generating (0.79 against 0.78 on League, 0.81 against 0.82 on World Cup) but recalls fewer of the play's carriers (0.81 against 0.87, 0.80 against 0.88), although it reads the whole play, after ten extra calls to simulate it; its text is no better (Table 1). JointDiff's plays are busier than ours, however (4.2 and 3.9 successive ball carriers per play on League and World Cup, repeats merged and loose-ball frames skipped, against 3.7 and 3.4; the real plays have 4.3 and 4.0), so we also narrate our own plays afterwards with the same model (Ours → Ours). On the same plays, co-generation keeps a smaller edge: recall 0.87 against 0.86 on League and 0.88 against 0.85 on World Cup, cF1_{gen} 0.81 against 0.79 and 0.83 against 0.81, Order 0.68 against 0.67 and 0.72 against 0.68. Most of the gap to JointDiff → Ours thus comes from the plays, and co-generation adds one to two points of coherence on top. The re-narrated text is close to the co-generated one on League (CIDEr 0.62 against 0.59) and poorer on World Cup (0.74 against 0.79).

Table 13: **Full coherence table** (ten calls, 20 samples). Carrier precision, recall, cF1 and order of the ball carriers a narration names, scored against the possession of its own generated play (cF1_{gen} and Order of Table 2). The reference chain against the real play is the ceiling; JointDiff → Ours narrates each of JointDiff's plays afterwards with our full-play model (Appendix A); Ours → Ours narrates our own generated plays with the same model, a control for play complexity. Narrations are event chains. Our row is shaded.

	League				*World Cup*			
Narration	Prec	Rec	cF1	Order	Prec	Rec	cF1	Order
JointDiff → Ours (full play)	0.79	0.81	0.78	0.65	0.81	0.80	0.78	0.65
Ours → Ours (full play)	0.78	0.86	0.79	0.67	0.82	0.85	0.81	0.68
Masked diffusion	0.66	0.49	0.54	0.41	0.74	0.75	0.72	0.61
Insertion flow	0.72	0.81	0.73	0.63	0.86	0.77	0.79	0.69
Insertion flow + LAG (ours)	0.78	0.87	0.81	0.68	0.82	0.88	0.83	0.72
Reference vs. real play (ceiling)	0.78	0.84	0.79	0.67	0.83	0.87	0.83	0.73

E.3 STEERING IN FULL

Extends Table 3 and Fig. 3 (§4.3). Table 14 extends Table 3 to three edits, and Fig. 5 repeats Fig. 3 on League. The windows of both figures are those with the highest adoption among the first 64 test windows in which both outcomes occur: best cases, whose frequencies over all windows are those of the tables. Even an edit to a random teammate, often far from the play, moves the ball to that player in half of the samples or more (50–68%, against 15–20% without narration), though less often than to the nearest teammate (69–91%), because the observed frames constrain who can receive. The old target keeps receiving in about a third of the samples partly because the swap is applied throughout the chain and moves it into any earlier clause of the new one (in Fig. 3, "#7 pass to #6" becomes "#3 pass to #6").

E.4 LAG STRENGTH

Extends §4.4. Table 15 varies the strength on both corpora. Narration quality and length rise with p_{LAG} and with a lower $\alpha_{\min}$, at a cost of one to two points of Acc_{gen}; the energy score is not monotonic in between, and the strongest setting, ours, has the best one on both corpora.

Scaling the insertion rate instead. A cheaper alternative multiplies the plain flow's predicted insertion rate at inference, leaving training untouched. On World Cup at ten calls, a factor of two raises CIDEr from 0.38 to 0.45 and the narration from 30% to 35% of the reference length, and a factor of four only to 0.46 and 37% (pF1 0.43 → 0.46, eF1 0.47, energy score 2.82 → 2.83), against CIDEr 0.79, 67%, pF1 0.53 and energy score 2.71 with LAG. League behaves alike: CIDEr 0.21 → 0.26 → 0.27 and length 26% to 30% for both factors (pF1 0.46 → 0.47, energy score 3.46 → 3.44), against 0.59, 66%, 0.53 and 3.25 with LAG. The gain saturates because the deficit is not a rate scale: the calibrated heads keep reading the short state as a short target (the frozen ratio of §3.4), which only training on behind-schedule states changes.

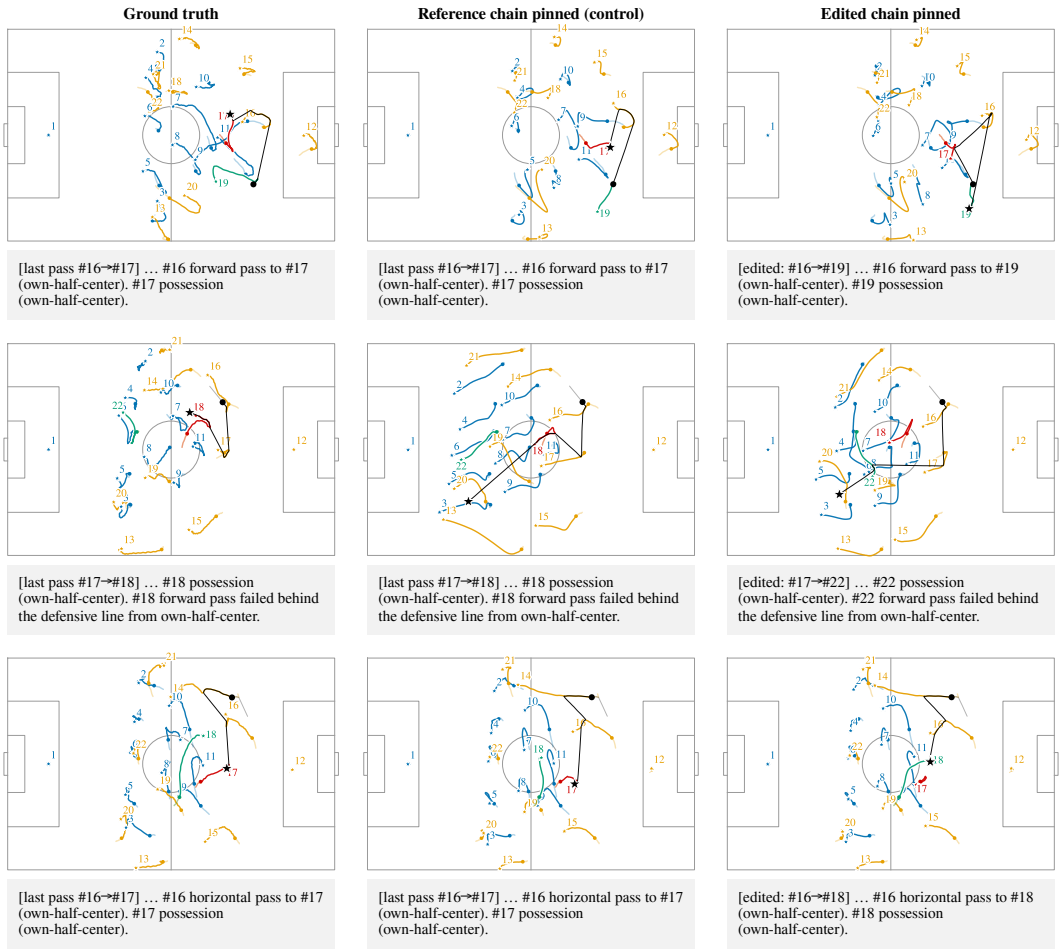


Figure 5: **Editing the story edits the play, League** (three test windows, last pass swapped with the nearest teammate; same protocol and sample selection as Fig. 3).

E.5 Sensitivity to the Sampling Budget

Extends §4.4 and effects (i) and (ii) of §3.4. From fifty to ten calls (Table 16), the plain insertion flow keeps 48–55% of its CIDEr, the collapse of §3.4, and ours 68–77% with pF1 flat. At five calls every insertion model collapses, as the ceiling $2^5{=}32 < L$ of effect (i) implies for any model; masked diffusion, with no length to lose, keeps the largest share of its CIDEr there and is the least-bad narrator on pF1, but always with the poorest trajectories. Average sADE moves by at most 0.24 m across budgets except for masked diffusion on League (0.71 m): the budget is mainly a *text-stream* sensitivity. Fifty calls do not lengthen our narration (66% and 67% of the reference length at ten calls, 61% and 67% at fifty).

More levels and calls. To test whether more calls alone close the gap, we retrain the plain insertion flow and ours on World Cup with $S{=}200$ levels and sample the first 512 test windows at $K{=}50$, 100 and 200. The plain flow saturates (CIDEr 0.99, 0.96 and 0.97, pF1 0.49–0.50, eF1 0.54) while LAG keeps its lead (CIDEr 1.31–1.32, pF1 0.53, eF1 0.58 at every budget), and our $S{=}50$ model at ten calls (0.98, 0.54, 0.52 on the same windows) already matches the plain flow's CIDEr and pF1 at a hundred; CIDEr's document frequencies are computed on the evaluated windows, so these subset values are not comparable with Table 1. The $S{=}200$ retrains keep the $S{=}50$ noise schedule without rescaling it, so we evaluate them at $K{\geq}50$ only and compare their narrations.

Table 14: **Full steering table** (ten calls, 20 samples). The edit of Table 3 and Fig. 3 for three edits of the reference event chain: the first pass or the last pass swapped with a random teammate, and the last pass with the nearest teammate. Text stream as in Table 3: JointDiff + Guid reads the chain as a prompt, ours has it pinned. Columns: share of samples in which the old target (the original receiver) and the new target hold the ball, share in which the new target is the first receiver after the passer, and minimum ball distance to each; n counts the windows whose edited pass is generated (League / World Cup). JointDiff + Guid was evaluated with the edited prompt on the nearest-teammate edit only.

		League					*World Cup*				
		receives the ball			min. dist. (m)		receives the ball			min. dist. (m)	
Model	Text stream	old	new	new first	old	new	old	new	new first	old	new
First pass, random teammate (n=3,901/2,458 windows)											
JointDiff	No narration	0.67	0.15	0.03	3.9	16.8	0.69	0.16	0.04	3.4	16.2
JointDiff + Guid	Reference prompt	0.99	0.13	0.01	0.2	17.6	0.98	0.13	0.01	0.3	16.9
	Own narration	0.66	0.14	0.03	4.2	17.7	0.69	0.15	0.04	3.6	16.9
Ours	Reference pinned	0.99	0.12	0.01	0.2	17.8	0.95	0.14	0.02	0.7	16.6
	Edited pinned	0.58	0.53	0.26	3.8	11.6	0.57	0.50	0.21	4.2	11.0
Last pass, random teammate (n=3,100/1,824 windows)											
JointDiff	No narration	0.40	0.19	0.06	8.3	15.5	0.41	0.20	0.06	9.0	15.0
JointDiff + Guid	Reference prompt	0.98	0.11	0.00	0.2	17.5	0.80	0.12	0.00	1.9	17.1
	Own narration	0.39	0.18	0.06	8.9	16.4	0.41	0.19	0.07	9.3	15.8
Ours	Reference pinned	0.97	0.11	0.00	0.3	17.8	0.78	0.14	0.01	2.4	16.7
	Edited pinned	0.34	0.68	0.45	7.1	8.1	0.29	0.55	0.30	9.3	7.9
Last pass, nearest teammate (n=3,100/1,824 windows)											
JointDiff	No narration	0.40	0.29	0.09	8.3	9.0	0.41	0.28	0.09	9.0	9.6
JointDiff + Guid	Reference prompt	0.98	0.19	0.01	0.2	8.2	0.80	0.19	0.01	1.9	8.8
	Edited prompt	0.28	0.96	0.65	5.6	0.4	0.27	0.77	0.52	7.3	2.3
	Own narration	0.39	0.26	0.09	8.9	9.9	0.41	0.27	0.10	9.3	10.1
Ours	Reference pinned	0.97	0.19	0.01	0.3	8.4	0.78	0.22	0.03	2.4	8.6
	Edited pinned	0.30	0.91	0.61	5.9	1.1	0.34	0.69	0.39	7.1	3.3

Table 15: **LAG strength** (ten calls, 20 samples), mean±std over the seeds counted in the last column. Length: narration length in words relative to the reference (for the two single-seed World Cup settings, from a re-sample of the same checkpoint). Our rows are shaded.

	LAG setting	CIDEr	pF1	length	sADE_{avg}	ES	Acc_{gen}	seeds
League	no LAG	0.21	0.455	26%	5.73	3.46	0.952	1
	p_{LAG}=0.5, $\alpha_{\min}$=0.3	0.45±0.02	0.518±0.001	53%	5.61±0.01	3.30±0.04	0.945±0.004	2
	p_{LAG}=0.75, $\alpha_{\min}$=0.3	0.46	0.514	53%	5.69	3.32	0.949	1
	p_{LAG}=1, $\alpha_{\min}$=0.3	0.49	0.518	55%	5.60	3.28	0.946	1
	p_{LAG}=1, $\alpha_{\min}$=0.1 (ours)	0.57±0.02	0.528±0.004	64%	5.65±0.06	3.25±0.01	0.939±0.001	3
World Cup	no LAG	0.40±0.02	0.446±0.007	31%	4.82±0.03	2.86±0.06	0.961±0.002	5
	p_{LAG}=0.25, $\alpha_{\min}$=0.3	0.56	0.479	42%	4.65	2.81	0.950	1
	p_{LAG}=0.5, $\alpha_{\min}$=0.3	0.58±0.06	0.485±0.015	44%	4.90±0.12	2.92±0.14	0.949±0.004	5
	p_{LAG}=0.5, $\alpha_{\min}$=0.1	0.62	0.488	49%	4.81	2.85	0.946	1
	p_{LAG}=0.75, $\alpha_{\min}$=0.3	0.69±0.02	0.507±0.005	52%	4.74±0.08	2.79±0.05	0.947±0.004	5
	p_{LAG}=1, $\alpha_{\min}$=0.3	0.75±0.05	0.517±0.005	57%	4.82±0.09	2.83±0.07	0.943±0.003	5
	p_{LAG}=1, $\alpha_{\min}$=0.1 (ours)	0.79±0.04	0.524±0.007	65%	4.82±0.10	2.77±0.05	0.939±0.002	5

Table 16: **Sampling budget** (20 samples; the runs of Table 1). CIDEr, BERTScore, pF1 and sADE_{avg} at 50, 10 and 5 calls, i.e. skip $\zeta = 1, 5, 10$ (50, 11 and 6 network evaluations; Appendix A). Our rows are shaded.

		CIDEr			BERTScore			pF1			sADE_{avg}		
	Model (calls)	50	10	5	50	10	5	50	10	5	50	10	5
League	Masked diffusion	0.167	0.112	0.073	0.897	0.880	0.860	0.457	0.470	0.461	6.53	6.14	5.82
	Insertion flow	0.447	0.213	0.083	0.915	0.897	0.861	0.499	0.455	0.329	5.81	5.73	5.60
	Insertion flow + LAG (ours)	0.759	0.587	0.076	0.921	0.912	0.838	0.529	0.532	0.353	5.67	5.59	5.47
World Cup	Masked diffusion	0.740	0.600	0.370	0.930	0.921	0.905	0.448	0.467	0.473	5.27	5.27	5.51
	Insertion flow	0.682	0.375	0.138	0.924	0.911	0.884	0.473	0.435	0.333	4.87	4.82	4.73
	Insertion flow + LAG (ours)	1.148	0.785	0.071	0.934	0.925	0.860	0.528	0.526	0.366	4.77	4.71	4.65

E.6 QUALITATIVE SAMPLES

Extends §4.4. **Supplementary video.** It animates the sampler with the generated possessor highlighted at every call (a ring when no one holds the ball): one play denoised in all three streams, as in Fig. 1; the collapse mechanism of Fig. 2; the League samples of Fig. 7 with and without LAG; and a steering edit as in §4.3.

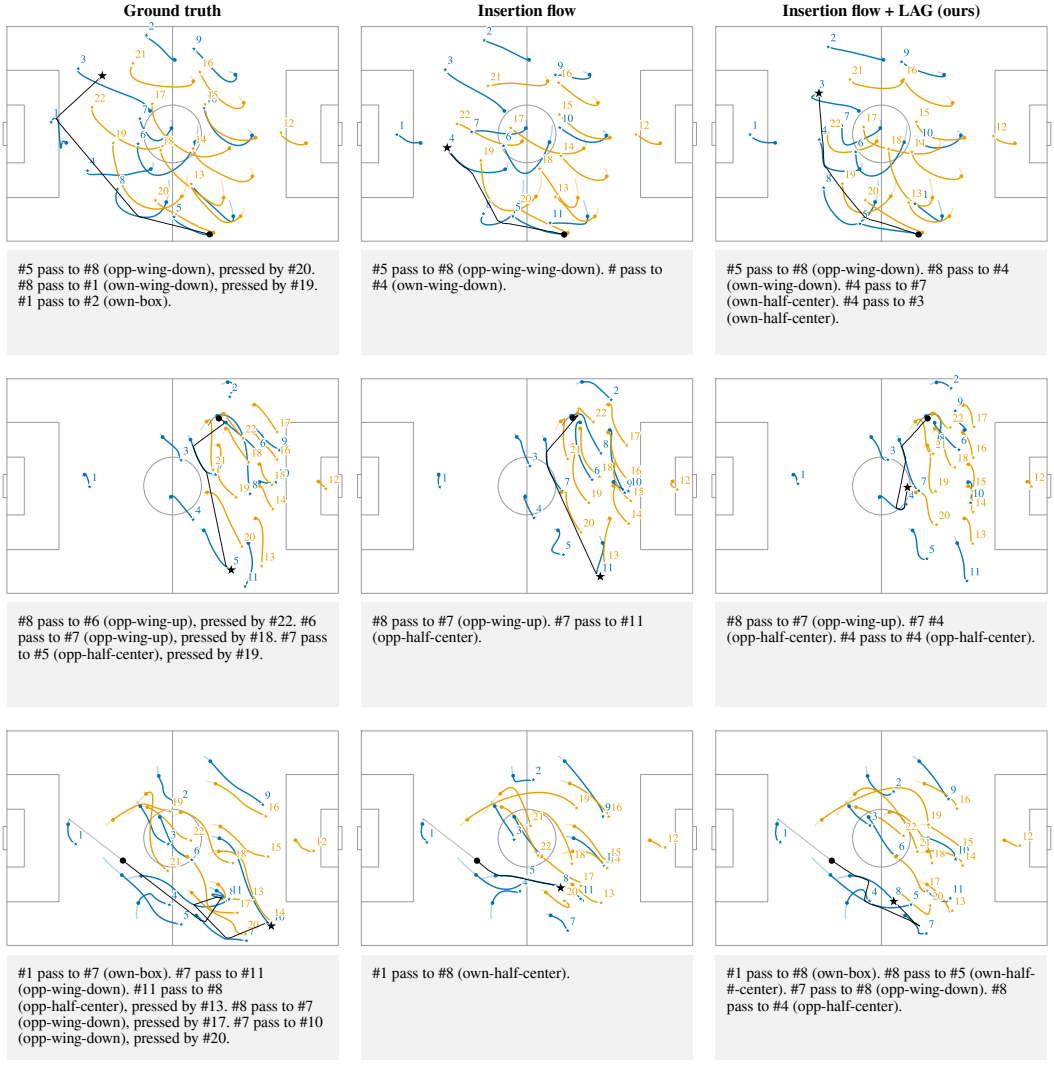


Figure 6: **Qualitative samples on World Cup** (three test windows; ten calls (K=10), 20 samples, the sample closest to the ground truth in scene ADE is shown). Left: ground truth and its event chain. Middle: insertion flow without LAG: the narration stops after one or two events and the ball path stays short (29–61 m against 56–84 m in the real plays). Right: with LAG the narration runs through more of the play and the ball travels farther in two of the three windows (41–62 m). Faded trails are the ten observed frames, • the last observed frame, ⋆ the end of the generated path, numbers the agent slots referenced in the text.

Fig. 6 shows the coupling of §4.3 window by window: the same three World Cup test windows generated with and without LAG, with the event chain each sample carries. Fig. 7 repeats it on League, whose chains name pressers, duels and failed passes: with LAG the narration runs through more of the play (four or five events against one or two), names the reference pressers in all three windows, and the ball travels farther (32 against 16 m, 45 against 37 m, 29 against 14 m).

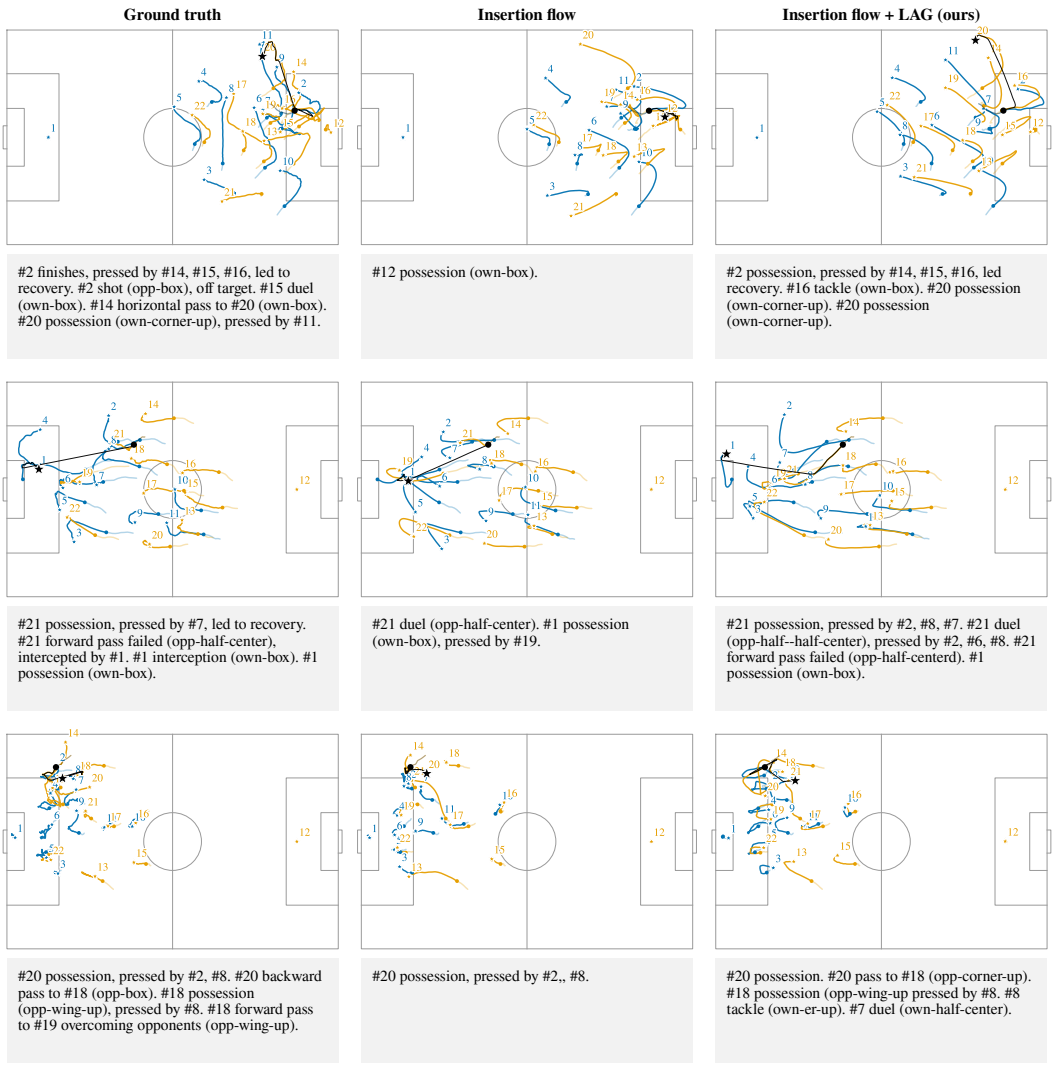


Figure 7: **Qualitative samples on League** (three test windows with a pressed possession in the reference; ten calls, 20 samples, the sample closest to the ground truth in scene ADE is shown). Left: ground truth and its event chain. Middle: insertion flow without LAG: the narration stops after one or two events. Right: with LAG it narrates four or five events, including the reference pressers, a tackle or a failed pass. Narrations are shown as generated, glitches included. Drawing conventions as in Fig. 6.

F Limitations and Future Work

Insertion-only flows cannot revise a committed token, and since the narration at most doubles per call, five-call sampling is limited to narrations of at most 32 tokens (§3.4); edit operations and multi-token insertion are natural extensions that would lift both. The masked baseline's canvas is sized for the longest chains, so most of it is padding; a length-predicted canvas or a remasking sampler could narrow its gap. LAG is tested inside the joint model on short event chains; a text-only flow and reveal schedules that respect the doubling limit would separate it from the trajectory coupling. Both corpora are football, while the framework itself is domain-agnostic: applying it wherever continuous dynamics and discrete descriptions co-evolve, with the narration as an editable plan for long rollouts or video synthesis, is the next step.

G STATEMENTS

G.1 LLM USAGE

Large language models were used in two ways. (1) To refine the writing and grammar of the manuscript, and to check that the numbers quoted in the text match the generated tables. (2) As a documented stage of the League caption pipeline (Appendix B.2), where a locally served Qwen3-32B (Yang et al., 2025a) rewrites each rule-based dense caption into prose under a prompt that requires every player, zone and outcome to be preserved verbatim and forbids adding information; World Cup narrations are produced by the rule-based grammar alone, with no LLM in the loop. The research questions, the model, the experimental design, the implementation and the interpretation of the results are the authors', who take full responsibility for the content.

G.2 ETHICS

This work uses trajectory data representing human agents. Both corpora are anonymized to team slots ordered by tactical role, with no names, jersey numbers, club or match identifiers, and the narrations refer to players only by those slots; the released League tier carries no absolute clock and hashed recording ids (Appendix B.2). The World Cup corpus is public. The intended applications, tactical analysis and multi-agent simulation, pose minimal ethical risk; we note that a model that narrates what it simulates is also easier to audit than one that does not.