Latent Space Steering for Controllable Rare Pedestrian Trajectory Generation

Ananya Arvind
Computer Science & Engineering
University of California, Los Angeles
ananyaarvind@ucla.edu

Abstract

Simulation-based testing of autonomous vehicles depends critically on diverse, high-fidelity scenario coverage, yet real-world pedestrian trajectory datasets are dominated by routine, low-activity motion, leaving the tail-event behaviors most relevant for safety assessment severely underrepresented. We investigate latent space steering as a lightweight, retraining-free approach to high-fidelity tail-event simulation: at inference time, we add a learned risk-direction vector to a trajectory encoder's latent representation to controllably shift generated pedestrian behaviors toward rare, high-activity modes without training any additional generative components. Using the Stanford Drone Dataset (SDD), we train and compare two sequential architectures—an LSTM and a Transformer trajectory predictor—and show both substantially outperform a constant-velocity baseline (LSTM: -48.0% ADE, Transformer: -58.3% ADE on the held-out test set). Linear probing of the learned latent spaces reveals a striking $2.55\times$ gap in how linearly each model encodes behavioral risk: Transformer $R^2 = 0.808$ vs. LSTM $R^2 = 0.316$. Structured latent steering outperforms random perturbation by $+0.159$ risk units (Transformer) and $+0.141$ (LSTM) at steering magnitude $\alpha = 1.5$, while maintaining physical plausibility above 94% throughout. Our central findings propose latent space steering as a lightweight, retraining-free method for controllable rare-behavior generation in AV simulation, and introduce latent probing R^2 as a complementary evaluation metric that captures representation quality invisible to standard ADE/FDE benchmarks. Our behavioral risk score is a kinematic activity proxy derived from speed, acceleration, and turn rate; extending this to scene-aware or learned danger scores is an important direction for future work.

1. Introduction

On-road testing of autonomous vehicles (AVs) is prohibitively costly and dangerous, making simulation an essential substrate for both evaluation and closed-loop training. A central challenge in simulation fidelity is scenario diversity: systems must be tested against the full distribution of agent behaviors, including rare tail events such as sudden pedestrian accelerations, sharp direction reversals, and erratic crossing patterns that are critical for safety assessment but underrepresented in real-world data [1]. The Stanford Drone Dataset (SDD) [2], one of the largest aerial pedestrian trajectory collections, exemplifies this long-tail imbalance: of 782,033 sliding-window trajectory clips extracted at 30 FPS, the high-activity behavioral modes most relevant for AV stress-testing remain a small minority.

Existing approaches to synthesizing safety-critical scenarios include adversarial trajectory generation [3] and diffusion-based motion models [4], but these typically require training additional generative components from scratch—adding significant complexity to the simulation pipeline. We instead ask a more targeted question: *does the latent space of an already-trained trajectory predictor encode behavioral risk as a linear structure, and if so, can we exploit this structure to generate rare, high-activity tail-event behaviors at inference time, with no retraining?*

This approach—**latent space steering**—draws inspiration from representation engineering in large language models [5, 6], where semantic concepts correspond to approximately linear directions in activation space. Translating this idea to trajectory prediction offers a lightweight, plug-in mechanism for enriching AV simulation with diverse rare behaviors: the steered trajectories could serve directly as challenging pedestrian agents in closed-loop simulation environments, providing physically plausible but behaviorally extreme scenarios without additional data collection or model training.

We directly compare two backbone architectures—**LSTM** and **Transformer**—answering two questions: (i) *which architecture better encodes behavioral risk as a linear structure in its latent space?* and (ii) *can latent steering controllably generate rare tail-event behaviors for AV simulation without retraining?*

Contributions.

- We train and benchmark LSTM and Transformer trajectory predictors on SDD against a constant-velocity baseline, reporting ADE and FDE on a held-out test set as a measure of prediction quality.
- We introduce a **latent probing** framework—linear R^2 regression from latent vectors to behavioral risk scores—revealing a $2.55\times$ representation-quality gap between architectures that is invisible to standard ADE/FDE metrics.
- We design a PCA-whitened mean-difference steering vector and demonstrate it outperforms random perturbation in generating high-risk trajectories while maintaining physical plausibility above 94%.
- We provide mechanistic insight connecting attention vs. recurrent compression to the geometric structure of each model's latent space, offering practical guidance for selecting trajectory prediction backbones in simulation pipelines.

This paper is intentionally exploratory: we treat the risk score as a kinematic activity proxy rather than a validated danger metric, and present the steering framework as a proof-of-concept whose effectiveness is demonstrated to hold independently of the specific score definition used.

2. Related Work

Pedestrian trajectory prediction. Social force models [7] captured pedestrian dynamics via hand-crafted physics. Social LSTM [8] replaced this with learned social pooling over recurrent hidden states, while Transformer-based predictors [9] demonstrated strong benchmark performance via self-attention over the full observation sequence. These works optimize primarily for ADE/FDE on held-out test sets. We study both architectures but focus on a dimension standard benchmarks ignore: representation quality for downstream controllable generation in simulation.

Safety-critical and tail-event scenario generation. A core challenge in AV simulation is generating rare but plausible scenarios for safety evaluation [1]. Rempe et al. [10] learn a traffic prior to synthesize accident-prone driving scenarios; Gu et al. [11] use diffusion for stochastic trajectory synthesis. These approaches train dedicated generative models. Our approach is complementary and lighter-weight: we exploit the internal latent structure of an existing trajectory predictor to steer toward tail-event behaviors without any additional training, making it easy to integrate into existing simulation pipelines.

Representation engineering and latent steering. Zou et al. [6] showed that semantic concepts in LLMs correspond to linear directions in activation space, enabling behavioral control via mean-difference vectors—a technique known as representation engineering. We are the first to apply this paradigm to trajectory prediction latent spaces, demonstrating its utility for generating rare agent behaviors in AV simulation.

Critically, our finding is not that Transformers outperform LSTMs on ADE/FDE (this is well-established) but that the two architectures differ fundamentally in the geometric structure of their latent spaces in ways that only become apparent when probing for downstream controllability.

3. Methods

3.1. Dataset and Preprocessing

We process all 60 annotation files from SDD across 8 scenes (bookstore, coupa, deathCircle, gates, hyang, little, nexus, quad). Bounding-box centers are converted to meters ($1\,\text{px} = 0.0375\,\text{m}$) and velocities estimated via finite differences at 30 FPS. A sliding window with $T_{\text{obs}} = 15$ frames (0.5 s), $T_{\text{pred}} = 25$ frames (0.83 s), stride $= 5$ yields **782,033** trajectory windows. Each is normalized to an agent-centric frame: the last observed position is translated to the origin and axes are rotated so the last observed velocity aligns with $+x$.

Splits. Windows are labeled *normal* (risk < 0.85, $n =$ 479,840) or *rare* (risk ≥ 0.85, $n = 302{,}193$), where rare denotes the top tail of our behavioral activity score rather than a ground-truth danger label. Normal windows are split 70/15/15 into train/val/test (335,888 / 71,976 / 71,976). The rare set is held out entirely and never seen during training, ensuring that any rare trajectories generated via steering cannot be attributed to direct memorization so that the model can generalize beyond its training distribution.

3.2. Behavioral Risk Score

We define a composite behavioral activity score $r \in [0, 1]$:

$$r = 0.35\, r_{\text{speed}} + 0.35\, r_{\text{accel}} + 0.30\, r_{\text{turn}} \tag{1}$$

where $r_{\text{speed}} = \min(v_{\max}/3.5, 1)$, $r_{\text{accel}} = \min(a_{\max}/3.0, 1)$, $r_{\text{turn}} = \min(\omega_{\max}/3.0, 1)$. Speed and acceleration are weighted equally (0.35) as primary activity indicators; turn rate is slightly lower (0.30) as sharp turns are more common in naturalistic motion. A score near zero corresponds to slow, straight-line pedestrian motion; a score near one corresponds to fast, erratic motion involving high speed, sudden acceleration, or sharp directional changes. This score is a proxy for behavioral unusualness rather than a ground-truth danger metric.

This score is intentionally kinematic: it encodes behavioral unusualness relative to the dataset distribution based on observable motion properties alone, without reference to scene geometry, other agents, or semantic context. Scene-aware or interaction-aware risk definitions e.g., time-to-collision with respect to nearby agents, or proximity to con-

flict zones in an HD map would produce a more ecologically valid danger metric. However, for the goal of latent space steering, a kinematic proxy is sufficient: we need only a score that is (a) computable from the same inputs the model observes, (b) linearly predictable from the learned latent representations, and (c) capable of separating the training distribution into high- and low-activity modes. We confirm all three properties hold empirically (§4.2). Replacing this score with a learned or scene-aware alternative is a concrete direction for future work (§5).

3.3. Model Architectures

Both models take 4D state vectors $[x, y, v_x, v_y]$ over T_{obs} timesteps and predict 2D positions $[x, y]$ over T_{pred} timesteps via an identical MLP decoder. Both are trained with MSE loss, since trajectory prediction is a continuous regression task where MSE directly optimizes for displacement error—the same quantity reported by ADE and FDE.

LSTM. Two stacked LSTM layers (hidden dim 128, dropout 0.2) process the observation autoregressively. The final hidden state $h \in \mathbb{R}^{128}$ is projected to $z \in \mathbb{R}^{64}$ via Linear+Tanh. (≈145K parameters.)

Transformer. Four self-attention layers ($d_{\text{model}} = 128$, 4 heads, FFN 256, dropout 0.2) with sinusoidal positional encodings process the full observation in parallel. The output is mean-pooled and projected to $z \in \mathbb{R}^{64}$ via Linear+Tanh. (≈320K parameters.)

Shared decoder. Both models use an identical MLP decoder that maps the latent vector $z \in \mathbb{R}^{64}$ directly to all future positions in a single forward pass: Linear(64→128) → ReLU → Dropout(0.2) → Linear(128→128) → ReLU → Linear(128→50), where the 50-dimensional output is reshaped to $(25, 2)$ representing the (x, y) coordinates of all $T_{\text{pred}} = 25$ future timesteps simultaneously. This non-autoregressive design means the entire predicted trajectory is decoded from a single latent vector, which is what makes latent steering effective: shifting z by αw at inference time modifies every future timestep in a coordinated way, rather than requiring sequential intervention.

Both models are trained end-to-end with the Adam optimizer (learning rate 1×10^{-3}, batch size 128) for up to 50 epochs, with early stopping based on validation ADE to select the best checkpoint. Training was conducted on a single NVIDIA T4 GPU via Google Colab, taking approximately 45 minutes per model.

3.4. Latent Probing

We fit ridge regression from $\{z_i\}$ to $\{r_i\}$ on the full validation set ($n = 71{,}976$) and report R^2. High R^2 indicates risk is encoded as an approximately linear function of z—a necessary condition for effective linear steering.

3.5. Steering Vector

We evaluate two methods for deriving the steering vector w:

1. **Mean-Difference (MD):** The difference between centroids of high-risk and low-risk latent clusters in a PCA-whitened space.
2. **Linear Probe (LP):** The normalized weight vector from the Ridge regression probe ($risk \sim z$).

In our results, we find MD vectors produce higher physical plausibility than LP weights, likely because the mean-difference is less susceptible to overfitting on high-variance noise dimensions in the latent space. Below details the steps used to compute the steering vector.

1. Extract $\{z_i\}$ over the validation set; apply PCA whitening (zero mean, unit variance along principal components) to prevent high-variance dimensions from dominating.

2. Split: $\mathcal{Z}_{\text{hi}}$ = top 10% by risk ($n = 7{,}308$); $\mathcal{Z}_{\text{lo}}$ = bottom 10% ($n = 22{,}955$).

3. $\tilde{w} = \text{mean}(\mathcal{Z}_{\text{hi}}) - \text{mean}(\mathcal{Z}_{\text{lo}})$ in whitened space; project back and ℓ_2-normalize.

4. Auto-correct sign: negate if negatively correlated with risk, so $\alpha > 0$ always increases risk.

We empirically evaluated percentile thresholds of 5%, 10%, and 20% for defining these clusters, finding that the 10% threshold provided the most stable steering directions by balancing sample size with the extremity of the behavior.

At inference:

$$z' = \text{clip}(z + \alpha \cdot w,\ -0.99,\ +0.99) \tag{2}$$

We sweep $\alpha \in [0, 1.5]$ and compare structured steering against a random-direction baseline ($z + \alpha \cdot w_{\text{rand}}$), each evaluated over 300 randomly sampled validation trajectories.

3.6. Physical Plausibility

A key requirement for steered trajectories to be useful in AV simulation is that they remain physically realistic; a trajectory that violates basic locomotion constraints is not a valid test case, regardless of its risk score. We therefore evaluate physical plausibility alongside risk to confirm that steering operates within the learned motion manifold rather than pushing predictions into physically impossible regions of the output space.

A generated trajectory is defined as *plausible* if its maximum instantaneous speed does not exceed 3.5 m/s, the empirical 99.5th percentile of unsteered model predictions on the validation set (observed maximum: 2.84 m/s). This threshold corresponds approximately to a fast human run, and was chosen data-driven rather than hand-tuned: trajectories below it are consistent with the natural speed distribution the model

Table 1. Test-set prediction accuracy.

Method	ADE↓	FDE↓	ΔADE	Params
Const. Velocity	0.137	0.265	N/A	N/A
LSTM	0.071	0.135	−48.0%	145K
Transformer	**0.057**	**0.106**	−58.3%	320K

learned during training, while trajectories above it indicate the decoder has been pushed beyond its reliable operating range.

We deliberately do not apply frame-level acceleration or heading-change constraints. Our MLP decoder regresses all T_{pred} timesteps simultaneously from a single latent vector, meaning consecutive output frames are not generated autoregressively and do not respect inter-frame dynamics by construction. As a result, empirical inter-frame heading changes are large even for unsteered predictions (p75: 174°), making such constraints architectural artifacts rather than physical violations. Speed-based plausibility is therefore the most reliable single indicator of whether a steered trajectory remains within the model's learned distribution, and serves as a conservative lower bound on physical realism.

4. Results

4.1. Prediction Accuracy

Table 1 reports ADE and FDE on the held-out test set, where ADE measures the mean Euclidean distance between predicted and ground-truth positions across all T_{pred} timesteps, and FDE measures this distance at the final timestep only. ADE captures how well a model tracks the overall trajectory shape, while FDE indicates whether the model has learned meaningful long-horizon dynamics. Both models substantially outperform constant-velocity extrapolation; the Transformer achieves ADE = 0.057 m—outperforming the LSTM by 19.7% on ADE and 21.4% on FDE (Figure 1).

Our results also compare favorably to goal-conditioned architectures like PECNet [12], which reported a benchmark ADE of 9.96 pixels (≈0.37m) on the same Stanford Drone Dataset. This provides further evidence that our Transformer and LSTM models are capturing high-fidelity motion dynamics consistent with top-performing models in the literature.

4.2. Latent Space Structure

Figure 2 shows PCA projections of each model's latent space, where each point represents a trajectory window encoded to $z \in \mathbb{R}^{64}$ and projected to 2D. PCA finds directions of maximum variance, so the distribution reveals whether risk varies smoothly along a linear direction—a necessary condition for effective steering.

The LSTM collapses to a near-1D manifold (PC1: 99.8%, PC2: 0.2%)—a curved ribbon where all trajectories are compressed onto a single axis, leaving little geometric structure for a steering vector to exploit. The Transformer is genuinely 2D (PC1: 79.5%, PC2: 14.2%) with a visible directional risk axis. The apparent scatter in the Transformer plot is not noise—it reflects the model using its full 64D space to encode diverse trajectory properties (speed, curvature, turn rate, etc.) that cannot all be captured in 2D. Crucially, even within the apparent cloud, a clear risk gradient is visible along PC1: points on the left of the projection are predominantly green (low risk) and points on the right are predominantly red (high risk), confirming that risk varies smoothly and directionally across the latent space. The scatter along PC2 represents other trajectory properties orthogonal to risk— evidence of a richer, more expressive representation rather than disorganization. When projected to 2D this structure appears diffuse, but full-space linear probing confirms it is highly organized: **Transformer** $R^2 = 0.808$ **vs. LSTM** $R^2 = 0.316$—a $\mathbf{2.55}\times$ gap despite comparable prediction accuracy.

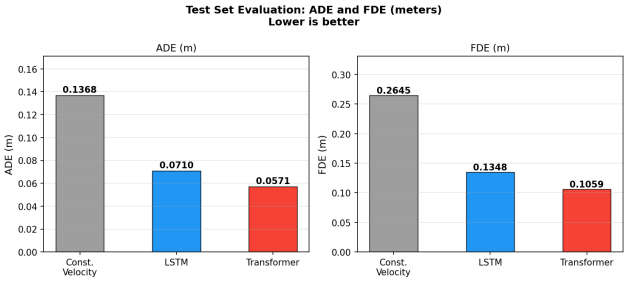

Figure 1. Test-set ADE and FDE. Both models substantially outperform constant-velocity extrapolation.

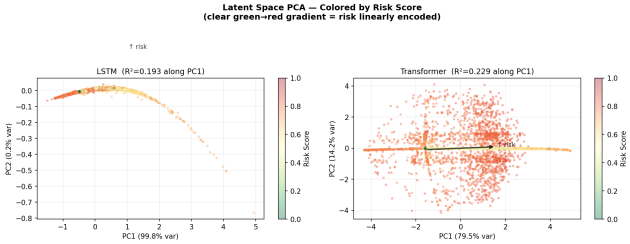

Figure 2. PCA projections of validation latent spaces colored by risk score (green = low, red = high). **Left (LSTM):** Near-1D manifold collapse (PC2 = 0.2% variance). **Right (Transformer):** Genuinely 2D structure (PC2 = 14.2%) with a clear linear risk direction (arrow). R^2 in panel titles reflects variance along PC1 only; full 64D probing yields LSTM $R^2 = 0.316$, Transformer $R^2 = 0.808$.

4.3. Steering Results

Figure 4 sweeps $\alpha \in [0, 1.5]$ over 300 validation trajectories. Four findings emerge:

(1) Risk increases monotonically. LSTM reaches mean risk 0.897 at $\alpha = 1.5$; Transformer reaches 0.798.

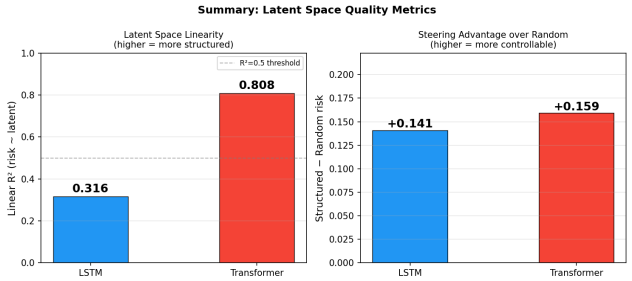

Figure 3. **Left:** Latent probing R^2. **Right:** Steering advantage over random baseline at $\alpha = 1.5$.

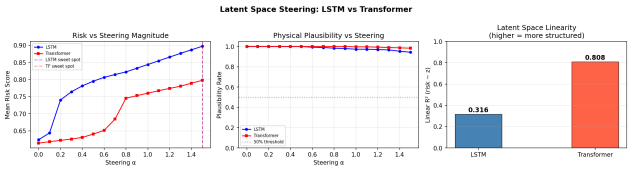

Figure 4. Latent steering results. *Left:* Mean generated risk vs. α; structured (solid) vs. random (dashed). *Center:* Plausibility stays $> 94\%$ throughout. *Right:* R^2 comparison.

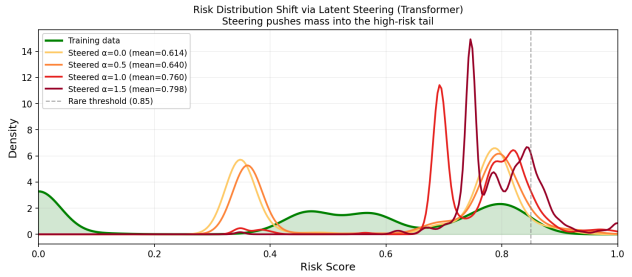

Figure 5. KDE of generated risk at increasing α (Transformer). Distributions narrow rather than simply translate, indicating the steering vector targets a specific behavioral mode rather than adding isotropic noise.

(2) Structured $>$ random for both models. LSTM: $+0.141$ advantage; Transformer: $+0.159$. These margins confirm the steering vectors encode semantically meaningful directions.

(3) Transformer shows a sigmoid-like transition at $\alpha \approx 0.8$, indicating a more concentrated risk direction consistent with its higher R^2.

(4) Physical plausibility maintained throughout: LSTM 94.3%, Transformer 98.3% at $\alpha = 1.5$.

4.4. Risk Distribution Shift

Figure 5 shows kernel density estimates of generated risk scores at $\alpha \in \{0, 0.5, 1.0, 1.5\}$ for the Transformer, illustrating how the output distribution evolves as steering magnitude increases.

At $\alpha = 0$ (no steering), the distribution peaks near 0.70–

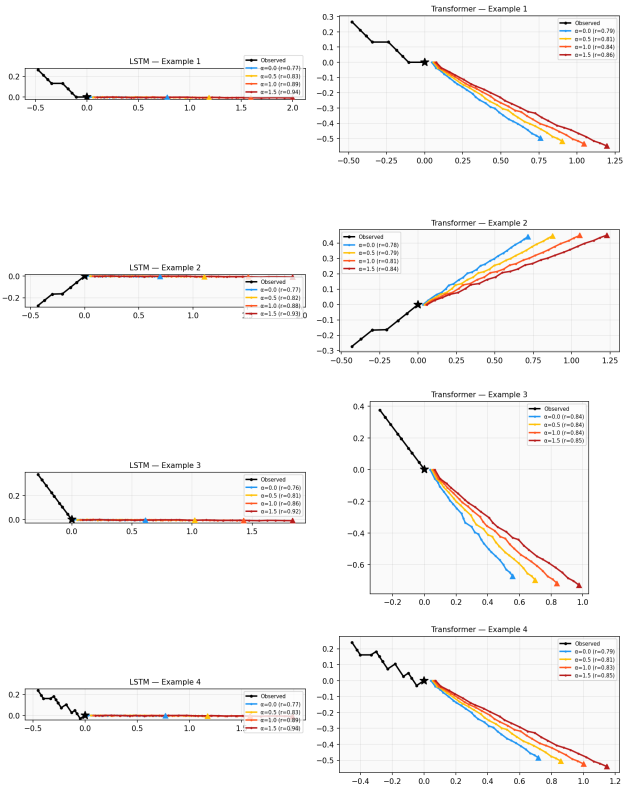

Figure 6. Steered trajectory examples. **Black:** observed history. **Blue→red:** predicted futures at $\alpha \in \{0.0, 0.5, 1.0, 1.5\}$. LSTM (left): speed-dominated. Transformer (right): geometrically diverse.

0.75 with a long left tail representing the majority of normal, low-activity trajectories in the validation set. As α increases, two distinct effects emerge. First, the distribution *shifts rightward*, with progressively more mass crossing the rare threshold at 0.85 by $\alpha = 1.5$, a substantial portion of generated trajectories exceed this threshold, confirming that steering successfully elevates behavioral activity. Second, and more importantly, the distribution *narrows* as α increases rather than simply translating. This narrowing is a strong indicator that the steering vector is encoding a semantically specific direction in latent space: it is consistently pushing trajectories toward a particular behavioral mode (high speed, high curvature, or sharp directional change) rather than adding diffuse, isotropic perturbations that would broaden or flatten the distribution. The fact that the Transformer's steering vector produces this narrowing effect while the LSTM's does not to the same degree is consistent with its higher R^2: a more linearly organized latent space yields a more precise steering direction, and a more precise direction produces a more concentrated output distribution.

4.5. Qualitative Analysis

Figure 6 shows steered trajectory examples for both models at $\alpha \in \{0.0, 0.5, 1.0, 1.5\}$, with the observed history shown in black and predicted futures color-coded from blue ($\alpha = 0$, unsteered) to red ($\alpha = 1.5$, maximum steering). The color progression makes it easy to trace how the predicted future changes as steering magnitude increases.

For the **LSTM**, the steered trajectories are dominated by a single behavioral mode: as α increases, predictions extend further and faster along roughly the same heading as the unsteered baseline. The trajectory fan is narrow, with all steered outputs pointing in similar directions and differing mainly in speed and length. This is consistent with the LSTM's near-1D latent collapse since all behavioral information is compressed onto a single axis correlated with speed, steering that axis produces speed-dominated responses with little geometric variety.

For the **Transformer**, the steered trajectories are more diverse. At higher α, predictions not only accelerate but also curve more sharply, change direction, and exhibit higher-curvature paths that would not appear in the unsteered output. The trajectory fan is wider, with different steered samples exploring qualitatively distinct behavioral modes with some accelerating along the current heading, others turning sharply, reflecting the richer structure of the Transformer's 2D latent space.

This qualitative contrast directly reinforces the R^2 finding and shows that a Transformer-based steering approach can produce a wider variety of challenging pedestrian behaviors from a single inference-time intervention, providing more diverse coverage of the tail-event distribution than the LSTM's speed-dominated responses.

5. Discussion

Why does the Transformer encode risk more linearly? The LSTM's near-1D collapse results from sequential hidden-state compression: as the hidden state integrates 15 observation frames, instantaneous high-risk events (sudden acceleration peaks, sharp directional changes) are averaged into a smooth trajectory summary rather than preserved as distinct geometric features. The resulting latent manifold has limited dimensionality, making it difficult to extract a clean linear risk direction. The Transformer's self-attention processes the full observation in parallel and can allocate dedicated heads to different temporal events. A sudden speed spike at frame 10 of 15 can be directly attended to and reflected in the final representation, rather than being diluted by the preceding 9 frames. The resulting 2D structure (PC2: 14.2%) provides the geometric space needed for an effective steering direction.

Accuracy vs. representation quality. Our central practical finding is the dissociation between ADE/FDE and R^2. The LSTM and Transformer differ by only 19% in ADE yet differ by $2.55\times$ in latent R^2, and categorically in the diversity of their steering behavior.

This suggests that when trajectory prediction models are intended for downstream generation or simulation (not just prediction), **latent probing R^2 should be reported alongside ADE/FDE** as a standard metric. A model with low ADE but collapsed latent geometry cannot support controlled rare-behavior synthesis.

Why does LSTM achieve higher absolute steered risk? The LSTM reaches higher absolute steered risk (0.897 vs. 0.798 at $\alpha = 1.5$) despite weaker R^2. Because all information in the LSTM's latent space is compressed along a single axis, any perturbation strongly affects the dominant encoded feature (correlated with speed). The LSTM thus produces *higher* absolute risk but *less controlled* behavior: its steering advantage over random ($+0.141$) is smaller than the Transformer's ($+0.159$), and its trajectory responses are geometrically less diverse. This illustrates the distinction between *forcing* high risk and *controllably generating* it.

Limitations and future directions. Several limitations of the current work point toward concrete future directions.

Risk score calibration. Our composite behavioral activity score (Eq. 1) uses hand-crafted normalization constants and serves as a proxy for unusualness within SDD rather than a ground-truth danger metric validated against human perception or real incident data. As a result, a trajectory scoring high on our metric is not guaranteed to be dangerous in a meaningful sense but rather it is simply unusual relative to the dataset distribution. A natural and important next step is replacing this with a learned danger score. Concretely, this could take several forms: (1) training a regressor on near-miss event annotations from datasets such as DOTA or the Honda Research Institute Driving Dataset (HDD); (2) using time-to-collision or post-encroachment time computed against HD map lane boundaries as a scene-aware proxy; (3) leveraging human hazard-perception ratings from crowdsourced annotation studies. Each would produce a score that generalizes beyond kinematic unusualness and makes steered trajectories more directly actionable for safety evaluation. We view this as the most important next step for the framework. We emphasize that the latent steering methodology is independent of the specific risk score used to derive the steering vector: any score that can be linearly regressed from the latent representations can serve as a steering target. The limitations address the current score's fidelity, not the framework's validity.

Decoder architecture. The MLP decoder regresses all T_{pred} timesteps simultaneously from a single latent vec-

tor, which simplifies the steering mechanism but sacrifices inter-frame physical consistency. Under large α, this can produce trajectories with unrealistic instantaneous velocity changes between consecutive frames. Autoregressive decoders—which generate each timestep conditioned on previous outputs—or diffusion-based decoders would impose stronger temporal coherence and enable richer plausibility constraints beyond speed alone, such as acceleration limits and smooth curvature profiles.

Reinforcement learning and training diversity. Beyond evaluation, steered trajectories could serve as an adversarial training signal in RL-based AV policy learning. By injecting rare pedestrian behaviors into the training environment, the policy can be exposed to tail events at a much higher rate than passive data collection allows, potentially improving robustness to exactly the scenarios that cause real-world failures.

Multi-agent and social interactions. The current model treats each pedestrian independently, which limits the realism of generated rare scenarios. Many of the most safety-critical pedestrian behaviors involve agent-agent interactions—a group of pedestrians crossing simultaneously, a child running into traffic from behind another agent, or a near-collision between two crossing pedestrians. Extending to graph-based or social attention encoders would allow steering to generate coordinated rare interactions rather than isolated unusual trajectories.

Generalization. Our experiments are conducted on SDD alone. Whether the $2.55\times$ R^2 gap between LSTM and Transformer latent spaces generalizes to other trajectory datasets (ETH/UCY, nuScenes, Waymo Open Dataset) and other agent types (vehicles, cyclists) remains an open and important empirical question.

6. Conclusion

We presented a study of latent space steering for high-fidelity tail-event simulation of pedestrian behavior, addressing a core challenge in AV simulation: generating rare, safety-critical scenarios without expensive on-road data collection or training additional generative models. Comparing LSTM and Transformer trajectory predictors on SDD, we found both substantially outperform constant-velocity extrapolation (-48% and -58% ADE), but linear probing exposes a $2.55\times$ gap in representation quality ($R^2 = 0.316$ vs. 0.808) that is entirely invisible to ADE/FDE. Structured latent steering reliably shifts generated trajectories into the rare tail while maintaining physical plausibility above 94%, outperforming random perturbation by up to $+0.159$ risk units.

These results carry two practical messages for the AV simulation community. First, when selecting a trajectory prediction backbone for a simulation or generation pipeline, latent probing R^2 should be reported alongside ADE/FDE a model that predicts well but encodes behavior poorly cannot support controlled rare-event synthesis. Second, inference-time latent manipulation offers a lightweight, plug-in route to tail-event diversity in simulation, and this property holds for any steering target that can be linearly regressed from the latent space not just the kinematic activity proxy used here. Replacing our proxy with a scene-aware or learned danger score is the most direct path to making this framework production-ready for safety evaluation.

Acknowledgements. Experiments used Google Colab with NVIDIA T4 GPU access. Code is available at https://github.com/ananya-arv/latent-steering.

References

[1] X. Ji, L. Xue, Z. He, and X. Luo. Autonomous driving system testing via diversity-oriented driving scenario exploration. *ACM Transactions on Software Engineering and Methodology*, 2025. 1, 2

[2] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. Learning social etiquette: Human trajectory prediction in diverse social scenarios. In *ECCV*, 2016. 1

[3] S. Agarwal and S. P. Chinchali. Synthesizing adversarial visual scenarios for model-based robotic control. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2022. 1

[4] M. M. Mammen, Z. Kayatas, and D. Bestle. Generation of multiple types of driving scenarios with variational autoencoders for autonomous driving. *Future Transportation*, 5(4):159, 2025. 1

[5] L. Bartoszcze, S. Munshi, B. Sukidi, J. Yen, Z. Yang, D. Williams-King, L. Le, K. Asuzu, and C. Maple. Representation engineering for large-language models: Survey and research challenges. *arXiv preprint arXiv:2502.17601*, 2025. 1

[6] A. Zou, L. Phan, S. Chen, J. Campbell, P. Guo, R. Ren, A. Pan, X. Yin, M. Mazeika, and A. Li. Representation engineering: A top-down approach to AI transparency. *arXiv:2310.01405*, 2023. 1, 2

[7] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Physical Review E*, 51(5):4282–4286, 1995. 2

[8] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In *CVPR*, 2016. 2

[9] F. Giuliari, I. Hasan, M. Cristani, and F. Galasso. Transformer networks for trajectory forecasting. In *ICPR*, 2020. 2

[10] D. Rempe, J. Philion, L. Guibas, S. Fidler, and O. Litany. Generating useful accident-prone driving scenarios via a learned traffic prior. In *CVPR*, 2022. 2

[11] J. Gu, C. Sun, and Y. Zhao. Stochastic trajectory prediction via motion indeterminacy diffusion. In *CVPR*, 2022. 2

[12] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *CVPR*, 2020. 4