Accurate Forecasts Do Not Ensure Safe Decisions

Jaehyun Pyun [1] **Seunghun Moon** [2] **Suk-Ju Kang** [2]

Abstract

Pedestrian forecasting is still largely evaluated with retrospective best-of-K metrics such as minADE and minFDE, although planners must commit to safe actions before the future is observed. We introduce the Trajectory-Decision benchmark protocol (TrajD), a compact decision-centric stress test built on the TrajImpute missing-trajectory benchmark dataset. TrajD fixes the incomplete-history carrier, ego-action set, planner, ego-agent selection rule, and ground-truth decision metrics, then audits five stochastic sources–MoFlow, SingularTrajectory, TUTR, NMRF, and SocialGAN–through one forecast-to-decision pipeline. Source planning leaves substantial avoidable collision risk. As a reference action-risk alignment method, a lightweight two-fold out-of-fold (OOF) aligner leaves source samples unchanged but rescales planner action scores, reducing collision by about one third on average; a less conservative operating point partially recovers goal progress. Results show that forecasting benchmarks should report downstream safe decisions alongside geometric sample accuracy.

1. Introduction

Pedestrian-facing robots must make safe decisions in dense crowds with unpredictable interactions, yet trajectory prediction is still largely evaluated by best-of-K displacement metrics (Gupta et al., 2018). These metrics reward the existence of one accurate sample while ignoring how the remaining forecast distribution affects action selection. A planner cannot retroactively choose the best sample after observing the future; it must commit to an action under the full predictive distribution at inference time.

Partial observability makes this mismatch more severe. Occlusions and tracking losses distort velocity, heading, and interaction cues. A stochastic forecaster may still generate geometrically plausible samples, but its default weights or scores may no longer reflect action risk. We call this a *forecast-to-decision failure*: a forecast set can look acceptable under retrospective displacement metrics while inducing unsafe downstream actions.

To quantify this failure mode, we introduce the Trajectory-Decision benchmark protocol (TrajD), a decision-centric stress test built on the TrajImpute missing-trajectory benchmark dataset (Chib & Singh, 2024). TrajD fixes the missing-history carrier, ego-agent selection rule, candidate ego-action set, CVaR-style planner, source-cache interface, and ground-truth decision metrics. As shown in Figure 1, we audit five stochastic sources—MoFlow (Fu et al., 2025), SingularTrajectory (Bae et al., 2024), TUTR (Shi et al., 2023), NMRF (Fang et al., 2025), and SocialGAN (Gupta et al., 2018)—through the same forecast-to-decision pipeline: observed coordinate sequences and masks are converted into finite source inputs, the source exports $K = 20$ future samples, the planner selects one ego action, and that action is scored against ground-truth futures.

Forecasting errors generally increase under missing observations, but geometric degradation alone is not our main question: even under clean observations, NMRF attains the best clean minADE_{20} (0.19) yet induces a higher source-planning collision rate than SocialGAN (4.4% vs. 3.2%), whose clean minADE_{20} is 0.61. The OOF action-risk alignment method is a reference layer, not part of the metric definition.

TrajD does not replace geometric metrics: best-of-K displacement still measures coverage of plausible futures, while TrajD measures whether the resulting distribution supports the one action a planner must choose before the future is known. A detailed discussion of related work is provided in Appendix A.

- We introduce the **Trajectory-Decision benchmark protocol (TrajD)**, a compact open-loop stress test for pedestrian forecasts under missing observations.
- We audit five stochastic forecasting sources through a shared forecast-to-decision interface, exposing a gap between geometric sample accuracy and downstream

[1]Department of Artificial Intelligence, Sogang University, Seoul, South Korea [2]Department of Electronic Engineering, Sogang University, Seoul, South Korea. Correspondence to: Suk-Ju Kang <sjkang@sogang.ac.kr>.

International Conference on Machine Learning, Forecast: Forecasting as a New Frontier of Intelligence, Seoul, South Korea.

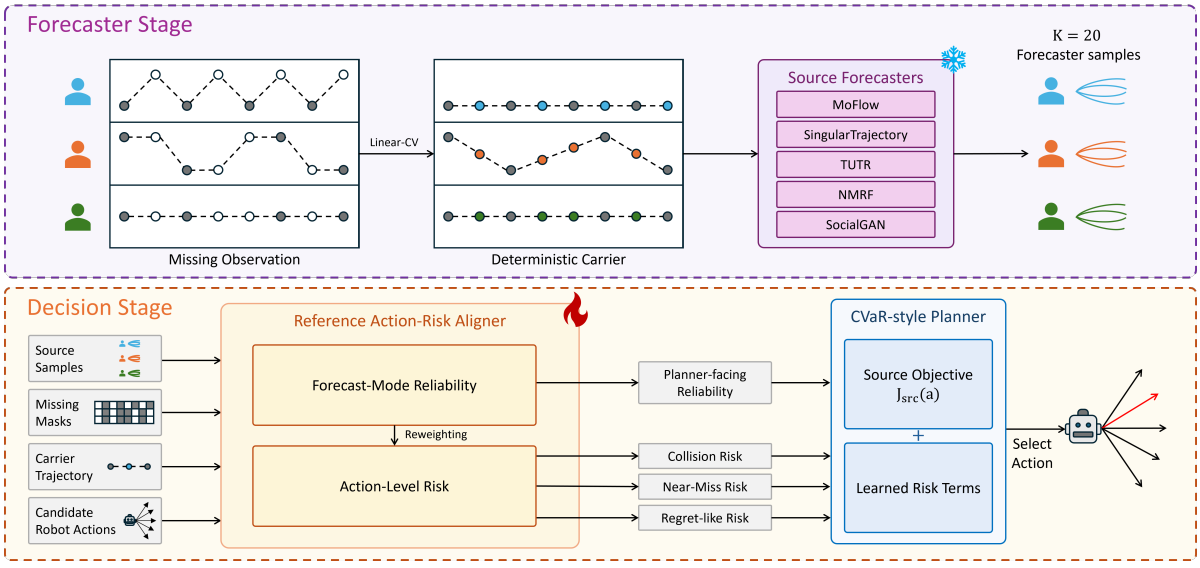

Figure 1. **TrajD evaluation pipeline and reference action-risk alignment method.** Inputs are observed pedestrian coordinate sequences with missing masks, not raw images. A deterministic Linear-CV carrier interfaces with frozen forecasters, which output $K = 20$ future coordinate samples. The decision stage scores 25 ego actions and outputs one selected action/rollout, evaluated only against ground-truth futures.

safe decisions.

- We provide a lightweight two-fold OOF reference action-risk alignment method that leaves source samples unchanged but learns planner-facing reliability and risk signals.

2. The TrajD Benchmark Protocol

2.1. Overview of the Forecast-to-Decision Pipeline

Figure 1 summarizes TrajD. Each episode starts from observed coordinate histories $O \in \mathbb{R}^{T_o \times N \times 2}$ and missing masks $M \in \{0,1\}^{T_o \times N}$. Frozen sources output $K = 20$ future coordinate samples; the decision stage maps them to one ego action $a^* \in \mathcal{A}$ and rollout, evaluated by ground-truth collision/progress. Sections 2.2–2.4 match the forecaster interface, decision metrics, and reference alignment block in Figure 1.

2.2. Episode Construction and Input/Output Interface

Episodes contain $T_o = 8$ observed and $T_f = 12$ future frames. We evaluate 10 conditions formed by 5 ETH/UCY datasets and two missingness regimes: Easy removes 0–4 observed frames, while Hard removes 4–7 observed frames. Missingness is applied only to past observations; future labels remain intact for evaluation.

Most off-the-shelf forecasters cannot consume NaN-valued histories, so TrajD uses a deterministic Linear-CV carrier (linear interpolation and constant-velocity extrapolation) to convert partial histories into finite coordinates. This non-learned carrier is not an imputation contribution; it exposes frozen sources to finite, perception-like coordinate noise as a controlled stress test while preserving the original missingness mask for the decision layer. Each source outputs $K = 20$ future samples. When calibrated joint scene scores are unavailable, we use a uniform default planning distribution; TUTR target-level scores are treated as diagnostic rather than calibrated joint likelihoods. A valid pedestrian is selected as an ego-controlled proxy agent, and invalid or oracle-infeasible cases are filtered before decision evaluation.

2.3. Fixed Action Space and Ground-Truth Decision Metrics

The robot selects from an action set $\mathcal{A}$ containing 25 semantic kinematics: stop, slow, forward, turns, and wait variants. Source planning selects $a^*_{\text{src}} = \arg\min_{a \in \mathcal{A}} J_{\text{src}}(a)$, with $J_{\text{src}}(a) = J_{\text{base}}(a; p_{\text{src}})$: a CVaR-style objective over predicted proximity plus terminal goal and freeze penalties. Thus the baseline is already risk-sensitive; TrajD tests whether its source distribution leads to safe actions. Appendix B gives the exact form and constants.

The selected action a^* is evaluated strictly against the **ground-truth** (Y^{GT}) futures, not the predicted ones. A **Collision** occurs if $\min_{t=1:T_f} \min_{j \neq r} \|x^{\text{ego}}_{a^*}(t) - Y^{\text{GT}}_{j,t}\|_2 < d_{\text{col}}$, where r is the ego-controlled proxy. **Progress** measures normalized terminal advancement toward the pseudo-goal, not total path length; see Appendix B for the exact formula. As a non-learned reference, we also report a validation-selected hand-written rule baseline that rejects

Table 1. Main decision results on TrajD. Values are condition-balanced averages over the ten Easy/Hard conditions. **Src.** denotes source planning, **Rule** is the validation-selected hand-written safety rule, **Hyb.** uses the collision-oriented reference OOF aligner, and **SRP** is the selected-risk-progress mode. All evaluated episodes are oracle-feasible: at least one candidate action in $\mathcal{A}$ is collision-free under the ground-truth future.

	Collision Rate (↓)				**Progress** (↑)			
Source Model	**Src.**	**Rule**	**Hyb.**	**SRP**	**Src.**	**Rule**	**Hyb.**	**SRP**
MoFlow	0.0297	0.0279	**0.0187**	0.0263	0.8033	0.7915	0.8277	**0.8451**
SingularTrajectory	0.0414	0.0328	**0.0267**	0.0313	0.7452	0.7358	0.8003	**0.8390**
TUTR	0.0328	0.0300	**0.0219**	0.0257	0.7596	0.7519	0.8128	**0.8377**
NMRF	0.0582	0.0540	**0.0411**	0.0453	**0.8871**	0.8555	0.8094	0.8600
SocialGAN	0.0431	0.0403	**0.0289**	0.0342	**0.8742**	0.8399	0.8329	0.8560

or penalizes actions with low predicted clearance and otherwise follows the same source planner.

2.4. Reference Action-Risk Alignment Method

TrajD is primarily a benchmark; this alignment method is a diagnostic reference baseline, not the main contribution. It never changes the frozen source samples. Given a source cache $\{\hat{Y}^{(k)}, s_k\}_{k=1}^{K}$, the preserved missing mask, the Linear-CV carrier, local scene statistics, and each candidate ego rollout, it estimates two planner-facing signals: reliability-aware weights over modes and action-level collision, near-miss, and regret-like risk values for $|\mathcal{A}| = 25$. Only planner action scores are modified.

Forecast-mode reliability. Best-of-K metrics reward whether at least one sample lands near the future, but planning depends on how the full predictive distribution supports or misleads action selection. The reliability module therefore takes source-level cues, missingness statistics, scene context, and pooled action context, and outputs a planner-facing mode distribution

$$p_{\mathrm{rel}}(k) = \mathrm{softmax}\left(\frac{s_k + \lambda_q q_k}{\tau}\right), \tag{1}$$

where s_k is the source-side mode score when available and a uniform log-score default otherwise, q_k is a learned reliability correction, and τ is a learned temperature. This stage estimates which forecast modes the planner should trust more under partial observation.

Action-level risk prediction and operating points. For each candidate action, the action-risk module evaluates action-sample clearance and geometry features against every forecast mode, aggregates them with p_{rel}, and predicts planner-facing penalties $\hat{r}_{\mathrm{col}}(a)$, $\hat{r}_{\mathrm{near}}(a)$, and $\hat{r}_{\mathrm{reg}}(a)$. Both components are lightweight MLP calibrators over compact tabular features, not sequence generators. To prevent leakage, they are trained on two-fold OOF source caches; learned parameters, collision-risk thresholds, and rule-baseline parameters are fit or selected using training/validation caches only. Test futures are never used for fitting the aligner, calibrating θ_{col}, tuning rule baselines, or selecting Hybrid/SRP operating points.

Hybrid (Collision-Oriented). Hybrid evaluates the same CVaR-style base planner with the learned mode distribution p_{rel} and adds the learned action-risk penalties:

$$J_{\mathrm{hyb}}(a) = J_{\mathrm{base}}(a; p_{\mathrm{rel}}) + \lambda_{\mathrm{col}}\hat{r}_{\mathrm{col}}(a) + \lambda_{\mathrm{near}}\hat{r}_{\mathrm{near}}(a) + \lambda_{\mathrm{reg}}\hat{r}_{\mathrm{reg}}(a) + \lambda_b\,\mathrm{softplus}\left(\frac{\hat{r}_{\mathrm{col}}(a) - \theta_{\mathrm{col}}}{\beta}\right). \tag{2}$$

The soft barrier is centered at a calibrated collision-risk threshold θ_{col}, making Hybrid explicitly collision-oriented rather than generically risk-averse.

Selected-Risk-Progress (SRP). SRP is a less conservative mode for the safety-efficiency trade-off. It first defines a safe action set using the learned collision-risk threshold,

$$\mathcal{A}_{\mathrm{safe}} = \{a \in \mathcal{A} \mid \hat{r}_{\mathrm{col}}(a) \le \theta_{\mathrm{col}}\}, \tag{3}$$

and, if non-empty, selects the progress-aware action within that set:

$$a^* = \arg\min_{a \in \mathcal{A}_{\mathrm{safe}}} \Big[J_{\mathrm{base}}(a; p_{\mathrm{rel}}) - \lambda_p\,\mathrm{Progress}(a) + \lambda_w\,\mathbf{1}[\mathrm{wait}(a)] + \lambda_s\,\mathbf{1}[\mathrm{stop}(a)]\Big]. \tag{4}$$

Here λ_p, λ_w, and λ_s are fixed planner weights for progress reward, wait penalty, and stop penalty. If the safe set is empty, SRP falls back to a softened risk-penalized objective. Hybrid prioritizes safety, whereas SRP recovers forward progress while remaining safer than source planning.

3. Experiments

3.1. Main Decision Results

Table 1 shows that source planning is not a sufficient decision rule under missing observations. Hybrid reduces collision for every source and achieves an average relative reduction of about one third compared with source planning. It also outperforms the validation-selected hand-written rule

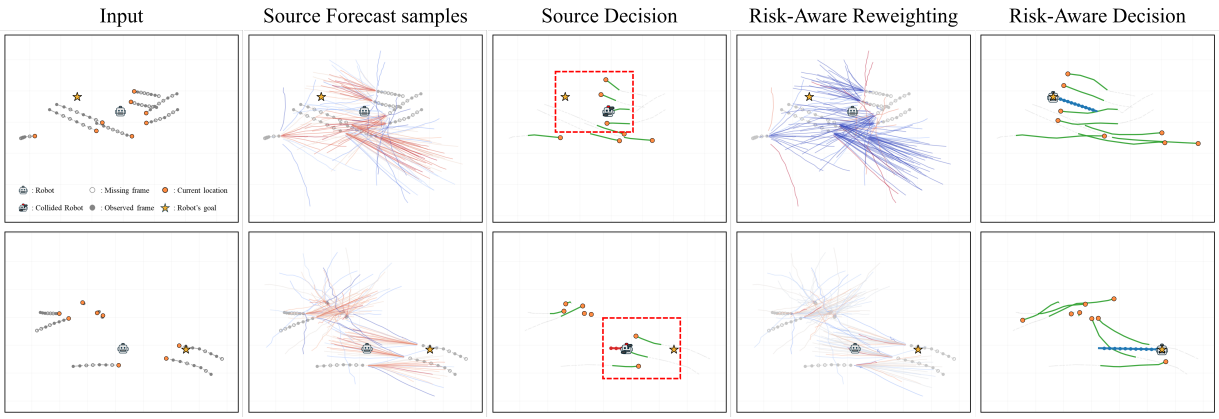

Figure 2. **Comparison of Source Planning and Risk-Aware Planning.** Each row shows the same episode under source planning (MoFlow) and the reference action-risk alignment layer. Green trajectories are ground-truth pedestrian futures. Forecast samples are identical across columns; only the decision rule changes. The risk-aware view colors the same source samples by the aligner's selected-action risk, where red indicates higher risk and blue indicates lower risk. Red bounding boxes denote collision states. Source planning selects a rollout that collides with the ground-truth future, whereas the risk-aware layer selects a safer action with positive progress.

baseline on collision for all five sources, indicating that the learned layer is not merely a trivial hard-coded distance filter.

For MoFlow (Fu et al., 2025), SingularTrajectory (Bae et al., 2024), and TUTR (Shi et al., 2023), Hybrid improves both collision and progress by down-weighting unreliable samples that would trigger spurious collision alarms and freezing. For NMRF (Fang et al., 2025) and SocialGAN (Gupta et al., 2018), Hybrid is more conservative: collision decreases, but progress is reduced. This behavior is intentional: Hybrid is the safety-first collision-oriented operating point. In these cases, SRP mitigates the conservative bias and recovers progress while remaining safer than source planning, providing a practical operating point on the safety-efficiency frontier. Figure 2 illustrates this mechanism qualitatively: the forecast samples are held fixed, while the selected ego action changes from colliding source planning to safer risk-aware planning. Additional qualitative examples can be found in Appendix C.

3.2. Clean-History Generalization

To test whether the aligner overfits to Linear-CV artifacts, we evaluate zero-shot on a no-missing clean-control split. This split also sharpens the forecast-to-decision gap: NMRF has the best clean minADE_{20} in Appendix Table 3 (0.19), whereas SocialGAN has a larger clean minADE_{20} (0.61), but their source-planning collision rates in Table 2 are reversed (0.0444 vs. 0.0318). Hybrid and SRP reduce collision for all sources on this diagnostic split, indicating that the failure mode is not merely a Linear-CV artifact. Because the clean-control split is self-constructed and not part of the official Easy/Hard benchmark, we use it only as a robustness check rather than as a main comparison.

Table 2. Zero-shot transfer to clean histories. The decision layer is trained on missing-observation OOF caches and evaluated without retraining on a self-constructed no-missing clean-control split. This diagnostic is excluded from the main Easy/Hard averages and tests whether collision reduction transfers beyond Linear-CV artifacts.

Source	Mode	Coll. ↓	Prog. ↑
MoFlow	Src.	0.0221	0.8065
MoFlow	Hyb.	**0.0158**	0.8340
MoFlow	SRP	0.0166	**0.8388**
SingularTraj.	Src.	0.0367	0.7402
SingularTraj.	Hyb.	**0.0171**	0.8115
SingularTraj.	SRP	0.0200	**0.8361**
NMRF	Src.	0.0444	**0.8904**
NMRF	Hyb.	**0.0263**	0.8327
NMRF	SRP	0.0307	0.8573
SocialGAN	Src.	0.0318	**0.8768**
SocialGAN	Hyb.	**0.0216**	0.8551
SocialGAN	SRP	0.0239	0.8614
TUTR	Src.	0.0237	0.7691
TUTR	Hyb.	**0.0180**	0.8461
TUTR	SRP	0.0190	**0.8535**

4. Conclusion

TrajD is a compact open-loop stress test for the gap between forecast samples and downstream planning. Forecast batches can look geometrically plausible yet induce unsafe actions; the reference OOF aligner shows that safer action scores are possible without changing source samples. TrajD encourages reporting both sample accuracy and decision safety. We will release the TrajD evaluation scripts, fixed action definitions, filtering code, source-cache interface, and evaluation splits.

Acknowledgements

This research was supported by the IITP(Institute of Information & Communications Technology Planning & Evaluation)-ITRC(Information Technology Research Center) grant funded by the Korea government(Ministry of Science and ICT) (IITP-2026-RS-2023-00260091, 50%) and Korea Institute for Advancement of Technology(KIAT) grant funded by the Korea Government(MOTIE) (P0020535, The Competency Development Program for Industry Specialist, 50%).

Impact Statement

This paper introduces a benchmark protocol and evaluation suite for safer Machine Learning in robotics and autonomous navigation. By highlighting the gap between geometric forecasting metrics and downstream decision safety under missing observations, our work encourages risk-aware and reliable AI systems. TrajD is an open-loop diagnostic benchmark and should not be interpreted as a certification of closed-loop deployment safety; deployment would still require closed-loop validation, hardware testing, and domain-specific safety analysis.

References

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., and Savarese, S. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 961–971, 2016.

Bae, I., Park, Y.-J., and Jeon, H.-G. Singulartrajectory: Universal trajectory predictor using diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17890–17901, 2024.

Bahari, M., Saadatnejad, S., Farsangi, A. A., Moosavi-Dezfooli, S.-M., and Alahi, A. Certified human trajectory prediction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12301–12311, 2025.

Chakraborty, K., Feng, Z., Veer, S., Sharma, A., Ding, W., Topan, S., Ivanovic, B., Pavone, M., and Bansal, S. Safety evaluation of motion plans using trajectory predictors as forward reachable set estimators. *IEEE Robotics and Automation Letters*, 2026.

Chib, P. S. and Singh, P. Pedestrian trajectory prediction with missing data: Datasets, imputation, and benchmarking. *Advances in Neural Information Processing Systems*, 37:124530–124546, 2024.

Chib, P. S., Nath, A., Kabra, P., Gupta, I., and Singh, P. Ms-tip: imputation aware pedestrian trajectory prediction. In *International Conference on Machine Learning*, pp. 8389–8402. PMLR, 2024.

Fang, Z., Hsu, D., and Lee, G. H. Neuralized markov random field for interaction-aware stochastic human trajectory prediction. In *The Thirteenth International Conference on Learning Representations*, 2025.

Fu, Y., Yan, Q., Wang, L., Li, K., and Liao, R. Moflow: One-step flow matching for human trajectory forecasting via implicit maximum likelihood estimation based distillation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 17282–17293, 2025.

Gu, T., Chen, G., Li, J., Lin, C., Rao, Y., Zhou, J., and Lu, J. Stochastic trajectory prediction via motion indeterminacy diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17113–17122, 2022.

Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., and Alahi, A. 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.

Liao, H., Kong, H., Li, Z., and Xu, C. Safecast: Risk-responsive motion forecasting for autonomous vehicles. *AI Open*, 2025.

Moller, K., Nyberg, T., Tumova, J., and Betz, J. Pedestrian-aware motion planning for autonomous driving in complex urban scenarios. *IEEE Open Journal of Intelligent Transportation Systems*, 7:365–378, 2026. doi: 10.1109/OJITS.2026.3655468.

Shi, L., Wang, L., Zhou, S., and Hua, G. Trajectory unified transformer for pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9675–9684, 2023.

A. Related Work

Human Trajectory Prediction. Early interaction-aware pedestrian forecasting models introduced social pooling mechanisms to capture multi-agent dependencies (Alahi et al., 2016). More recent methods commonly employ stochastic generators based on GANs (Gupta et al., 2018), Transformers (Shi et al., 2023), or diffusion models (Gu et al., 2022; Bae et al., 2024) to produce multimodal futures. Most of these methods, however, remain optimized and evaluated under retrospective best-of-K displacement metrics, leaving the downstream action-selection consequences of the full predictive distribution underexplored.

Missing Observations and TrajImpute. Real robotic perception pipelines frequently suffer from occlusion and tracking failures. TrajImpute (Chib & Singh, 2024) standardizes this setting by injecting controlled missingness into observed histories. Imputation-aware predictors such as MS-TIP (Chib et al., 2024) instead modify the forecasting model itself to reconstruct missing observations, while certified trajectory prediction studies robustness guarantees under severe corruption or incomplete inputs (Bahari et al., 2025). Our work does not propose a new imputation or certification architecture. Instead, it uses missingness as a controlled stress-test condition for decision making and asks whether forecast distributions that appear geometrically reasonable remain safe when coupled to downstream planning.

Adjacent Safety-Aware Forecasting and Planning. Related but less direct work in autonomous driving studies safety evaluation of motion plans via predictor-derived forward reachable sets (Chakraborty et al., 2026), risk-responsive motion forecasting (Liao et al., 2025), and pedestrian-aware motion planning with explicit safety-efficiency trade-offs (Moller et al., 2026). These directions are aligned in spirit with our emphasis on downstream safety, but they target integrated vehicle-side forecasting or planning stacks rather than a benchmark for pedestrian forecast-to-decision reliability under missing observations.

B. Forecasting Sanity Tables and Protocol Details

This appendix provides the full forecasting sanity tables and protocol details behind the TrajD evaluation. The purpose is twofold. First, the forecasting tables characterize the geometric behavior of each frozen source model under the clean ETH/UCY setting and under the TrajImpute Easy/Hard missing-observation regimes. Second, the protocol details clarify how TrajD converts partial-observation forecasting episodes into downstream decision episodes through fixed filtering, action construction, and safety-oriented metrics.

Importantly, the Clean, Easy, and Hard settings are not identical protocols. Clean corresponds to the canonical no-missing ETH/UCY forecasting evaluation, whereas Easy and Hard are TrajImpute-derived partial-observation evaluations that use the fixed Linear-CV carrier to convert missing observations into finite coordinates before decision evaluation. Therefore, non-monotonic rows should be interpreted as protocol and carrier effects rather than as evidence that missing observations improve forecasting accuracy.

B.1. Forecasting Sanity Results

Table 3. **Clean forecasting sanity:** minADE_{20} / minFDE_{20} (↓).

Model	ETH	Hotel	Univ	Zara1	Zara2	Average
MoFlow	0.40 / 0.57	0.11 / 0.17	0.23 / 0.39	0.15 / 0.26	0.12 / 0.22	0.20 / 0.32
SingularTrajectory	0.35 / 0.42	0.13 / 0.19	0.25 / 0.44	0.19 / 0.32	0.15 / 0.25	0.21 / 0.32
TUTR	0.40 / 0.61	0.11 / 0.18	0.23 / 0.42	0.18 / 0.34	0.13 / 0.25	0.21 / 0.36
NMRF	0.26 / 0.37	0.11 / 0.17	0.28 / 0.49	0.17 / 0.30	0.14 / 0.25	0.19 / 0.32
SocialGAN	0.87 / 1.62	0.67 / 1.37	0.76 / 1.52	0.35 / 0.68	0.42 / 0.84	0.61 / 1.21

The forecasting sanity tables show that missing observations generally worsen geometric prediction quality, especially under the Hard regime. The degradation is most pronounced for endpoint error, indicating that long-horizon prediction becomes substantially less reliable when only a small number of observed history frames are available. NMRF is particularly sensitive to the missing-observation carrier, while MoFlow, SingularTrajectory, and TUTR remain comparatively stronger under Easy but still degrade under Hard. These results are diagnostic: they characterize the frozen sources used by TrajD, but they are not used to claim that TrajD itself improves source forecasting accuracy.

Table 4. **Easy missing-observation forecasting sanity:** minADE_{20} / minFDE_{20} (↓).

Model	ETH	Hotel	Univ	Zara1	Zara2	Average
MoFlow	0.51 / 0.74	0.15 / 0.27	0.27 / 0.46	0.19 / 0.32	0.14 / 0.23	0.25 / 0.41
SingularTrajectory	0.40 / 0.61	0.14 / 0.22	0.29 / 0.49	0.22 / 0.39	0.16 / 0.28	0.24 / 0.40
TUTR	0.59 / 1.01	0.13 / 0.19	0.27 / 0.47	0.21 / 0.38	0.16 / 0.28	0.27 / 0.47
NMRF	1.45 / 2.31	0.57 / 0.84	0.67 / 1.30	0.52 / 1.08	0.85 / 1.69	0.81 / 1.44
SocialGAN	0.80 / 1.39	0.45 / 0.90	0.63 / 1.23	0.28 / 0.51	0.27 / 0.49	0.49 / 0.90

Table 5. **Hard missing-observation forecasting sanity:** minADE_{20} / minFDE_{20} (↓).

Model	ETH	Hotel	Univ	Zara1	Zara2	Average
MoFlow	0.78 / 1.04	0.45 / 0.64	0.46 / 0.72	0.33 / 0.47	0.27 / 0.39	0.46 / 0.65
SingularTrajectory	0.79 / 1.10	0.39 / 0.57	0.50 / 0.78	0.39 / 0.57	0.28 / 0.42	0.47 / 0.69
TUTR	0.86 / 1.18	0.48 / 0.72	0.49 / 0.77	0.50 / 0.72	0.29 / 0.43	0.53 / 0.77
NMRF	2.78 / 4.32	0.63 / 1.31	1.91 / 3.15	1.38 / 2.29	1.23 / 2.18	1.58 / 2.65
SocialGAN	1.14 / 1.60	0.93 / 1.54	1.12 / 1.86	0.69 / 1.09	0.62 / 0.99	0.90 / 1.42

B.2. TrajD Episode Construction and Protocol Statistics

TrajD turns TrajImpute forecasting episodes into decision episodes. Each episode contains observed histories $O \in \mathbb{R}^{T_o \times N \times 2}$, observation masks $M \in \{0,1\}^{T_o \times N}$, futures $Y \in \mathbb{R}^{T_f \times N \times 2}$, and valid-agent masks $A \in \{0,1\}^{N}$. Missing observations are converted into finite coordinates using the fixed Linear-CV carrier, while the original missingness mask is preserved for the decision layer. This allows the benchmark to evaluate downstream decision quality under controlled partial observation, rather than merely reporting forecasting error.

Table 6. **TrajD protocol statistics over the ten Easy/Hard conditions.** Valid episodes are retained after robot selection, initial-clearance filtering, neighbor filtering, and oracle-feasibility filtering.

Dataset	Difficulty	Episodes	Valid Episodes	Valid Rate	Missing Ratio	Robot Obs. Frames	Scene Density
ETH	Easy	350	185	52.9%	25.2%	6.0	1.8
ETH	Hard	280	116	41.4%	62.4%	3.0	1.8
Hotel	Easy	1,505	685	45.5%	25.1%	6.0	3.0
Hotel	Hard	1,204	406	33.7%	63.0%	3.0	2.9
Univ	Easy	4,735	4,612	97.4%	25.0%	6.0	24.6
Univ	Hard	3,788	2,769	73.1%	62.5%	3.0	24.6
Zara1	Easy	3,010	2,265	75.2%	25.0%	6.0	3.1
Zara1	Hard	2,408	1,378	57.2%	62.5%	3.0	3.1
Zara2	Easy	4,605	2,800	60.8%	25.2%	6.0	5.9
Zara2	Hard	3,684	1,721	46.7%	62.7%	3.0	5.9

The *Episodes* column reports the number of generated forecasting episodes before decision filtering. *Valid Episodes* reports the number retained after robot selection, initial-clearance filtering, neighbor filtering, and oracle-feasibility filtering. *Valid Rate* is the retained fraction of generated episodes. *Missing Ratio* is the average fraction of missing coordinates in the observed history. *Robot Obs. Frames* is the mean number of observed past frames available for the selected robot out of $T_o = 8$. *Scene Density* is the mean number of pedestrians in the episode.

The table shows a clear separation between the two missingness regimes. Easy conditions have an average missing ratio of approximately 25% and retain roughly six observed robot frames, whereas Hard conditions have an average missing ratio of approximately 62–63% and retain roughly three observed robot frames. The valid rate also drops consistently from Easy to Hard, as severe partial observation makes it more difficult for an episode to satisfy the kinematic, interaction, and oracle-feasibility requirements. Dense scenes such as Univ retain many decision-valid interaction episodes, whereas smaller or sparser scenes such as ETH and Hotel naturally yield lower valid rates after filtering.

Table 7. **Decision-valid filtering pipeline.** Each TrajD episode is retained only if it satisfies all filtering stages.

Stage	Filter	Role in the benchmark
1	Future validity	Ensures that the selected ego agent and relevant neighbors have valid future trajectories for ground-truth decision evaluation.
2	Initial clearance	Removes cases where the ego agent already starts in collision or near-overlap, which would make planner comparison ill-posed.
3	Ego motion range	Keeps only candidates whose observed speed lies within the eligible range $[v_{\min}, v_{\max}]$.
4	Interaction requirement	Requires at least one nearby pedestrian within the neighbor radius, avoiding trivial single-agent navigation cases.
5	Density cap	Removes overly crowded starts with more than $N_{\max}$ neighbors, where all candidate actions may become degenerate.
6	Oracle feasibility	Retains only episodes where at least one action in $\mathcal{A}$ is collision-free under the ground-truth future.

B.3. Decision-Valid Filtering

A robot candidate is accepted only if it passes all filtering stages. This filtering removes trivial non-interactive episodes as well as impossible dense-start cases where every planner would fail. The resulting benchmark is therefore a controlled decision testbed: it evaluates action selection under partial observation while ensuring that each retained episode has meaningful interaction structure and at least one feasible safe action.

B.4. Constants, Actions, and Metrics

Table 8. **Protocol constants, thresholds, and action-space summary.** Distances are in world-coordinate units.

Quantity	Value	Description
T_o / T_f	8 / 12	Observed history and forecast horizon.
K	20	Forecast samples retained per episode.
$\lvert\mathcal{A}\rvert$	25	Fixed candidate robot actions.
Base actions	9	Stop, slow/normal forward, and left/right turns of 15, 30, and 45 degrees.
Wait variants	16	Two-frame and four-frame wait-then-go variants of the non-stop base motions.
d_{col} / d_{near}	0.3 / 0.6	Collision and near-miss thresholds.
d_{safe} / d_{start}	0.8 / 0.8	Safety hinge radius and minimum initial clearance.
$v_{\min}$ / $v_{\max}$	0.05 / 2.5	Eligible ego speed range per observed step.
r_{nbr} / $N_{\max}$	4.0 / 8	Neighbor radius and maximum allowed neighbors.
α_{CVaR} / λ_g	0.2 / 0.2	CVaR tail mass and terminal goal weight.
λ_q / τ range	1.0 / [0.5, 3.0]	Reliability correction weight and learned temperature range.
λ_{cvar} / λ_{freeze}	1.0 / 0.05	CVaR cost and freeze penalty weights.
λ_{col} / λ_{near} / λ_{reg}	2.0 / 0.5 / 0.5	Hybrid learned-risk penalty weights.
λ_b / β	2.0 / 0.05	Hybrid soft-barrier weight and scale.
λ_p / λ_w / λ_s	0.2 / 0.05 / 0.1	SRP progress reward, wait penalty, and stop penalty.
θ_{col} / rule params	validation-calibrated	Learned-risk thresholds and rule parameters are selected before test evaluation.
ϵ_{prog}	10^{-6}	Stability constant in the progress denominator.

The nominal ego heading and speed are estimated from the last two finite carrier observations. Stop uses zero velocity, slow-forward uses $0.5v$, normal-forward uses v, 15° and 30° turns use $0.8v$, and 45° turns use $0.7v$. Wait variants keep the ego fixed for two or four frames and then execute the corresponding non-stop base action.

For each candidate action a and source forecast mode k, we define the predicted geometric proximity cost

$$c_{\text{geo}}(a,k) = \sum_{t=1}^{T_f} \max\big(0,\ d_{\text{safe}} - d_{a,k,t}\big), \tag{5}$$

where

$$d_{a,k,t} = \min_{j \neq r} \left\| x_a^{\text{ego}}(t) - \hat{Y}_{j,t}^{(k)} \right\|_2 . \tag{6}$$

Given a planner mode distribution p, we define the CVaR-style base objective

$$J_{\mathrm{base}}(a;p) = \lambda_{\mathrm{cvar}}\,\mathrm{CVaR}_{\alpha}\big(\{c_{\mathrm{geo}}(a,k)\}_{k=1}^{K};p\big) + \lambda_g\,\|x_a^{\mathrm{ego}}(T_f) - g\|_2 + \lambda_{\mathrm{freeze}}\,\mathbf{1}[\mathrm{freeze}(a)]. \tag{7}$$

The source/default planner is $J_{\mathrm{src}}(a) = J_{\mathrm{base}}(a;p_{\mathrm{src}})$, where p_{src} is score-based when source scores are available and otherwise uniform. Learned operating points use the same base objective with p_{rel}, then add or constrain actions using learned action-risk terms. This appendix form makes explicit that the source planner is already risk-sensitive before the learned aligner is applied.

The reported hand-written rule baseline is selected on validation episodes before test evaluation. Its main selected variant uses a low predicted-clearance filter,

$$d_{\mathrm{rule}}(a) = \mathrm{Quantile}_q\left(\left\{\min_{t,j\neq r}\left\|x_a^{\mathrm{ego}}(t) - \hat{Y}_{j,t}^{(k)}\right\|_2\right\}_{k=1}^{K}\right), \tag{8}$$

rejects actions with $d_{\mathrm{rule}}(a) < \rho$, and selects the lowest-J_{base} action among the remaining candidates. The quantile q, threshold ρ, and fallback behavior are validation-selected and fixed before test evaluation.

For any selected action a, the ground-truth minimum distance is

$$d_{\min}^{\mathrm{GT}}(a) = \min_{t}\min_{j\neq r}\left\|x_a^{\mathrm{ego}}(t) - Y_{j,t}^{\mathrm{GT}}\right\|_2. \tag{9}$$

Collision and near-miss labels are computed as

$$\mathrm{Collision}(a) = \mathbf{1}\left[d_{\min}^{\mathrm{GT}}(a) < d_{\mathrm{col}}\right], \qquad \mathrm{NearMiss}(a) = \mathbf{1}\left[d_{\min}^{\mathrm{GT}}(a) < d_{\mathrm{near}}\right]. \tag{10}$$

Progress measures terminal goal advancement rather than total path length. The pseudo-goal g is estimated only from the finite carrier observation history, not from the ground-truth future. Specifically, if $o_r(T_o)$ and $o_r(T_o-1)$ are the last two carrier positions of the robot, we set

$$g = o_r(T_o) + T_f\big(o_r(T_o) - o_r(T_o-1)\big), \tag{11}$$

with a tiny default forward displacement used only for numerically stationary histories. Let $x_a^{\mathrm{ego}}(0)$ denote the current robot position before executing action a, and let $x_a^{\mathrm{ego}}(T_f)$ be the final point of the selected action rollout. We compute

$$d_0 = \|x_a^{\mathrm{ego}}(0) - g\|_2, \qquad d_1 = \|x_a^{\mathrm{ego}}(T_f) - g\|_2, \tag{12}$$

$$\mathrm{Progress} = \frac{d_0 - d_1}{\max(d_0, \epsilon_{\mathrm{prog}})}. \tag{13}$$

Positive progress indicates that the selected action moves the robot closer to the pseudo-goal over the 4.8-second future horizon.

B.5. Aligner Feature Inputs and Supervision

The reference decision layer is an action-level module rather than a forecasting module. For each episode and candidate action, it consumes the frozen source cache, the preserved observation mask, the Linear-CV carrier observation, local scene statistics, and the ego rollout induced by that candidate action. At inference time it never receives ground-truth futures and never modifies the source forecast samples.

Feature groups. The aligner uses four groups of inputs. **Missingness features** summarize the preserved observation mask, including the overall missing ratio, robot missing ratio, neighbor missing ratio, and observation-gap statistics. **Source-cache features** summarize the frozen forecast set and optional source scores; when true joint scene scores are unavailable, this branch falls back to a uniform-score indicator. **Action-forecast geometry features** summarize how each candidate action interacts with the frozen source samples through predicted clearance, tail clearance, and source-side planning cost. **Action and scene features** summarize terminal goal distance, nominal progress, speed or turn category, wait or stop indicators, current neighbor count, current minimum distance, and local density.

Predicted action-sample geometry. The main geometric primitive is the predicted action-sample clearance

$$d^{\mathrm{pred}}_{e,a,k,t} = \min_{j \neq r} \left\| x^{\mathrm{ego}}_{e,a}(t) - \hat{Y}^{(k)}_{e,j,t} \right\|_2 , \tag{14}$$

from which the aligner derives compact action-level summaries such as minimum predicted clearance, mean predicted clearance, tail predicted clearance, source-side CVaR cost, and terminal progress. These quantities describe how risky a candidate action appears under the frozen forecast set before the future is revealed.

Action-risk supervision. Training supervision is defined at the action level using ground-truth futures. For each episode and candidate action, we compute the closest ground-truth clearance

$$d^{\mathrm{GT}}_{e,a} = \min_{t,j \neq r} \left\| x^{\mathrm{ego}}_{e,a}(t) - Y^{\mathrm{GT}}_{e,j,t} \right\|_2 . \tag{15}$$

Collision and near-miss labels are obtained by thresholding this clearance at d_{col} and d_{near}. We also define a ground-truth action cost

$$C^{\mathrm{GT}}(a) = \max(0, d_{\mathrm{safe}} - d^{\mathrm{GT}}_{e,a}) + \lambda_g \left\| x^{\mathrm{ego}}_{e,a}(T_f) - g_e \right\|_2 , \tag{16}$$

and use the nonnegative regret target

$$r^{\mathrm{GT}}_{\mathrm{reg}}(a) = C^{\mathrm{GT}}(a) - \min_{a' \in \mathcal{A}} C^{\mathrm{GT}}(a'). \tag{17}$$

The action-risk head is trained with a focal binary cross-entropy loss for collision, a binary cross-entropy loss for near-miss, a Smooth L1 regret loss, and a pairwise action-ranking loss. The reliability head is trained with KL and cross-entropy terms against planner-facing mode targets on held-out source predictions, plus mild entropy-matching and preservation regularizers. The aligner therefore learns planner-facing action penalties that rank actions according to safety under the frozen source forecasts. Ground-truth futures are used only to construct these supervision targets and evaluation metrics; they are never used as aligner inputs at test time.

Source Score Semantics and OOF Hygiene. MoFlow exposes logit or probability-like source scores; logits are converted to p_{src} by a softmax over the K modes, while probability-like scores are clipped to nonnegative values and renormalized. NMRF, SingularTrajectory, and SocialGAN do not expose a true joint scene likelihood and are therefore assigned $p_{\mathrm{src}}(k) = 1/K$. TUTR provides averaged target-level scores; these are retained as diagnostic input features but are not treated as calibrated joint scene probabilities in p_{src}. The learned decision layer is trained using two-fold out-of-fold source caches. These caches are generated inside the training/validation partitions: each source checkpoint predicts episodes not used to fit that checkpoint. After the decision layer, thresholds, and rule parameters are fixed, final evaluation uses frozen source predictions on held-out test episodes, and test labels are used only for reporting.

B.6. Risk Diagnostics and Clean-Control Sanity

The action-risk score is used as a planner-facing ranking signal rather than as a perfectly calibrated collision probability. Table 9 reports both discrimination and calibration diagnostics. The AUROC and AUPRC values indicate that the score provides useful action ranking, while the ECE values caution against interpreting the score as an exact probability.

Table 9. **Action-risk discrimination and calibration diagnostics.**

Source	ECE ↓	Brier ↓	AUROC ↑	AUPRC ↑
MoFlow	0.1320	0.0994	0.9153	0.6197
NMRF	0.2008	0.1365	0.8571	0.4641
SingularTrajectory	0.1667	0.1136	0.9093	0.5941
SocialGAN	0.1359	0.1068	0.8923	0.5568
TUTR	0.1552	0.1077	0.9119	0.6071

The clean-control split is self-constructed and diagnostic-only. It is constructed by applying the same ego-selection, neighbor, initial-clearance, and oracle-feasibility filters to no-missing ETH/UCY episodes. It is not part of the official TrajImpute Easy/Hard benchmark, is not used for training, is not used for validation, is not used for threshold calibration, and is not

included in the main condition-balanced averages. Its purpose is to verify that the decision layer behaves sensibly when missingness is removed.

For this reason, we report the aggregate clean-transfer result once in the main text as a robustness check on zero-shot generalization to no-missing histories (Table 2), rather than duplicating it as an appendix benchmark table. The clean-control evidence should therefore be interpreted as diagnostic support against carrier-artifact overfitting, not as an additional official benchmark condition.

C. Additional Qualitative Examples

For NMRF, SingularTrajectory, SocialGAN, and TUTR, the source planner does not provide per-sample forecast scores; accordingly, all forecast samples receive uniform source scores in the source-planning view.

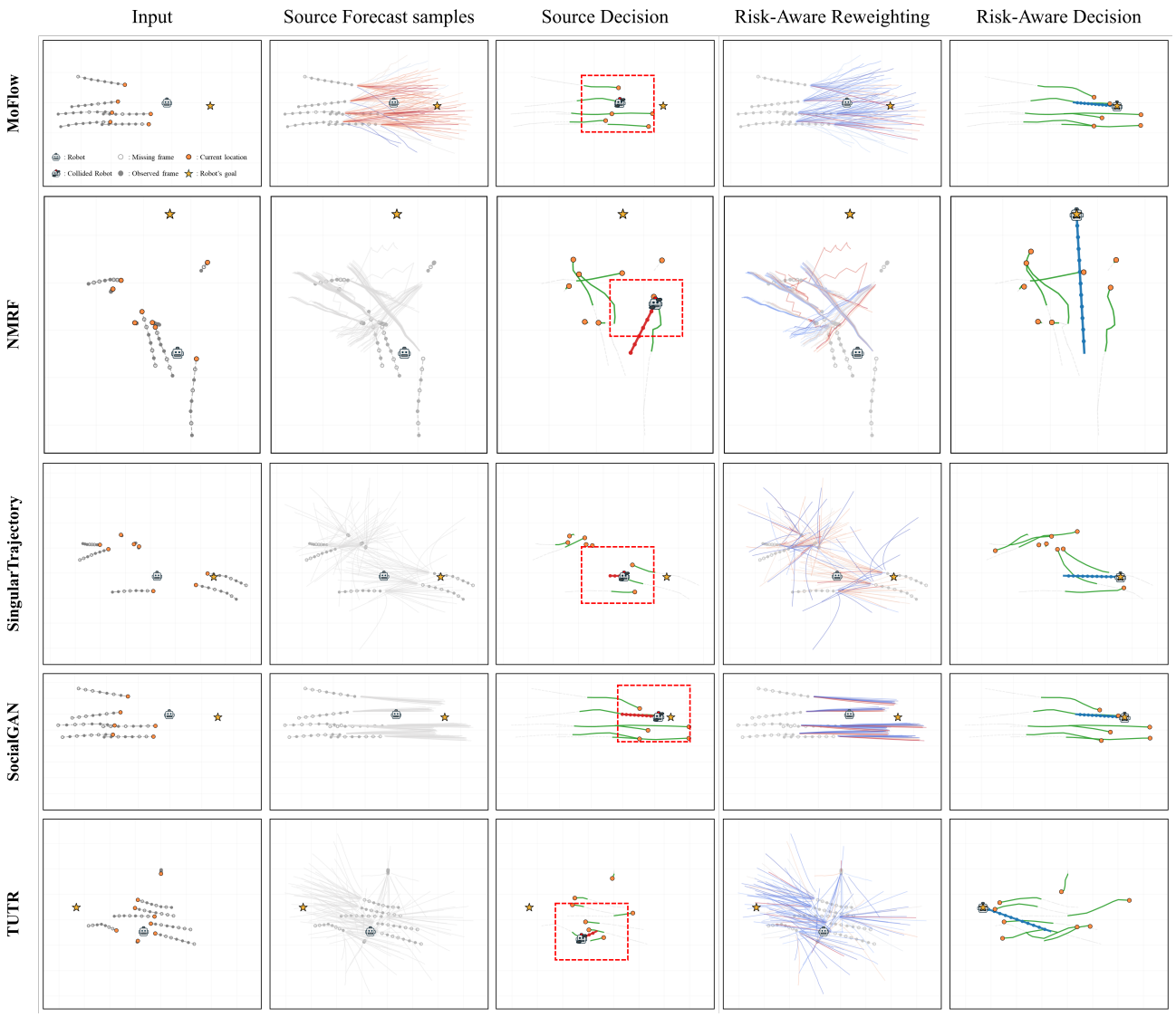

Figure 3. **Additional qualitative examples.** Each row compares source planning and the reference action-risk alignment layer on one representative held-out episode. Green trajectories denote ground-truth pedestrian futures. Forecast samples are identical across columns; only the decision rule changes. In the risk-aware view, the same source samples are colored by the aligner's selected-action risk, where red indicates higher risk and blue indicates lower risk. Red bounding boxes denote collision states. Source planning selects a rollout that collides with the ground-truth future, whereas the reference layer selects a safer action with positive progress.