Adapting SHARP to Heterogeneous Traffic for the HetroD Challenge

Alexander Prutsch Christian Fruhwirth-Reisinger David Schinagl Horst Possegger
Institute of Visual Computing, Graz University of Technology
{alexander.prutsch, reisinger, david.schinagl, possegger}@tugraz.at

Abstract

Behavior prediction research commonly uses large-scale motion forecasting benchmarks recorded in the US. In contrast to these car-centric datasets, European urban traffic is highly heterogeneous and includes a large share of vulnerable road users. Overhead drone recordings offer an effective way to capture such diverse scenes by densely tracking all agents within an area. The HetroD dataset [4] provides these recordings, and its associated challenge targets the joint simulation of numerous agents in complex situations like intersections. To tackle this challenge, we adapt SHARP [7], a strong streaming motion forecasting model, to the specific requirements of the task. We modify the model's output to fit the challenge format and fine-tune it on the provided training data. We also introduce three domain-specific refinements: map compliance is supervised across all predicted modes rather than only the best-fitting one, collisions are supervised against the surrounding traffic, and the scored agents of a training scene are sampled across agent types. A post-hoc module then reorders the rollouts of each agent to disentangle the remaining collisions. Our adaptation improves the challenge score from 0.57 zero-shot to 0.73 on the full validation split, resulting in a strong submission for the HetroD benchmark.

1. Introduction

Large-scale motion forecasting datasets [2, 3, 5, 8] are typically recorded in the US using instrumented vehicles. This data is often vehicle-centric, and the task aims to predict the future trajectories of a single focal agent or multiple agents. A small set of potential future trajectory modes is then scored against the corresponding ground-truth futures. European urban traffic is considerably more heterogeneous, as vehicles, two-wheelers, and pedestrians often share the same intersections and movement areas [1, 6]. Recording such scenes from an overhead drone allows for the dense tracking of all agents within an area, resulting in highly comprehensive data.

The HetroD Challenge [4] evaluates this specific setting using highly diverse data. A valid submission must provide 32 rollouts for every scored agent, and the evaluation metric combines kinematic realism, safety, consistency across agent types, and the spatial coverage of the rollout set. The kinematic term measures the likelihood of the simulated feature distribution under the ground truth, penalizing predictions that are either significantly smoother or rougher than the recorded motion.

While existing motion forecasting models provide a strong architectural baseline for the HetroD challenge, crucial adaptations to the specific task requirements are necessary. We adapt SHARP [7], a streaming motion forecasting model originally focusing on the Argoverse 2 (AV2) [8] multi-agent benchmark, to the HetroD challenge. First, we adjust the model's output to the challenge format, which requires 32 modes instead of six alongside an explicit heading prediction. Next, we align the training objective with the drone-recorded setting and supervise map compliance across all 32 modes rather than simply applying a winner-takes-all principle. We further supervise collisions against the surrounding traffic and balance the agent types among the scored agents of a training scene. Finally, we integrate a post-hoc collision refinement module to further improve the predictions.

In summary, our main contributions are:

- We adapt SHARP, a streaming motion forecasting model, to the specific requirements of heterogeneous, drone-recorded traffic in the HetroD dataset.
- We fine-tune our model on the HetroD dataset and introduce task-specific enhancements.
- Our experiments demonstrate that these additions, combined with fine-tuning on the challenge data, significantly improve overall performance.

2. Adapting SHARP to Heterogeneous Traffic

Our challenge submission builds upon the SHARP [7] Argoverse 2 multi-agent model, which predicts $K = 6$ trajectory modes of xy-coordinates over a 10 s horizon. We retain the encoder, the streaming context modules, and the decoder of SHARP, and adapt its prediction output to the challenge requirements. The model is then fine-tuned on the HetroD data, and we introduce domain-specific refinements in the training protocol. Furthermore, we employ a post-hoc collision disentanglement module to improve joint rollout consistency.

Problem Definition. A HetroD scenario [4] spans 91 time steps at 10 Hz, where the first 11 steps are observed and the goal is to predict the subsequent $T_f = 80$ steps. For every scored agent, the submission must provide $K = 32$ rollouts $F \in \mathbb{R}^{K \times N_a \times T_f \times 4}$ in global coordinates, consisting of the position (x, y, z) and the heading γ. Unlike standard motion forecasting evaluations, no probability scores are used. The challenge metric consists of a kinematic, a safety, a cross-type, and a coverage term. The kinematic term evaluates the simulated speed, acceleration, yaw rate, and yaw acceleration distributions, while the safety term couples the collision rate with the fraction of predicted positions inside the valid region. The cross-type term assesses consistency across agent types, and a coverage term rewards the spatial diversity of an agent's 32 rollouts.

2.1. Output Adaptations

Mode Expansion. To adapt SHARP from AV2 to the HetroD [4] output format, we first expand the number of output modes from $K = 6$ to $K = 32$. The query-based decoder enables this expansion simply by increasing the number of mode queries. To leverage the pretrained checkpoint, we retain the initial six queries and fill the remaining slots by perturbing the original modes with Gaussian noise. The additive noise is scaled using the mean pairwise distance between the six queries and distributed across the feature dimension.

Heading Prediction. While SHARP predicts only xy-coordinates, the challenge requires a heading for each state. We add a heading head to the global consistency module, following the same architecture as the standard position head. Rather than regression of the absolute heading per step, this head predicts a yaw *rate* that we integrate over the horizon starting from the last observed heading. This approach prevents the abrupt heading changes that an unconstrained per-step regression might produce.

2.2. Training Objectives

Partial Futures. In AV2, every scored agent is observed for the full horizon, so the standard objective masks invalid agents but not invalid time steps. In HetroD, 24% of the scored agents have tracks that end before the horizon, and the dataset pads these targets with zeros. Since a zero target in the agent-local frame denotes the current agent position, these padded steps must be masked per time step to ensure a correct loss computation. Without this masking, the objective trains the affected agents to return to their starting points once their tracks end, an artifact occurring in 49.6% of the predicted modes.

Heading Loss. Following standard motion forecasting regression losses, we add an additional loss $\mathcal{L}_{\text{yaw}}$ supervising the heading of the best-fitting mode. It applies a Smooth L1 loss to the residual between the predicted and recorded heading, wrapped to $(-\pi, \pi]$ to account for angle periodicity.

All-Mode Map Compliance. Standard training for motion forecasting models commonly uses a winner-takes-all strategy where only the best-matching mode receives a gradient to promote diverse likely trajectories. Because all 32 rollouts are scored equally in HetroD, this paradigm provides insufficient supervision; all modes should for instance comply with the map. To address this, we introduce a term that penalizes the mean distance by which a predicted footprint leaves the valid region for its agent type. This distance is retrieved from a precomputed signed distance field, and the penalty is applied equally across all 32 modes. We denote by $p_{i,t}^k$ the predicted global position of agent i in mode k at step t, and by d_c the signed distance to the valid region of agent type c, which is positive outside of it. The map compliance loss $\mathcal{L}_{\text{map}}$ is then formulated as

$$\mathcal{L}_{\text{map}} = \frac{1}{Z_{\text{map}}} \sum_{k,i,t} m_{i,t} \max\left(0, d_{c_i}\left(p_{i,t}^k\right)\right),$$

where $m_{i,t}$ masks the padded steps and Z_{map} normalizes by the number of valid predictions.

Collisions with Surrounding Traffic. A HetroD scene holds roughly 50 agents, but memory constraints limit us to supervising 16 scored agents per scene. Considering only collisions within the batch of scored agents, which are rare, is thus insufficient. To account for the remaining traffic, we treat the other agents in a scene as static obstacles along their recorded futures and retain the 16 nearest ones for every scored agent. Since each agent is predicted in its own local frame, we transform all footprints to global coordinates before computing their overlap. We approximate the shape of each agent using a set of circles aligned with the vehicle's heading, or a single circle for pedestrians, and then compute a pairwise collision loss $\mathcal{L}_{\text{coll}}$ between all scored agents and neighboring agents. For each collision, only the involved scored agent(s) receive a gradient. This also prevents the objective from simply pushing the whole scene apart.

Scored Agent Sampling. The subset of scored agents that receives supervision also determines which interactions the global consistency module observes during training. Sampling these agents from a spatial neighborhood provides realistic pairs, yet it under-represents the pedestrians of a scene. We therefore fill half of the sampling budget round-robin across the three agent types and the remaining half with the agents closest to a randomly drawn seed agent. This keeps a balanced set of types in every batch together with a genuine spatial neighborhood.

Final Objective. Our additional loss terms $\mathcal{L}_{\text{map}}$, $\mathcal{L}_{\text{coll}}$, and $\mathcal{L}_{\text{yaw}}$, are added to the scene-level loss of SHARP, which combines the winner-takes-all regression loss $\mathcal{L}_{\text{reg}}$ and the mode classification loss $\mathcal{L}_{\text{cls}}$, giving us

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{cls}} + \lambda_{\text{yaw}}\mathcal{L}_{\text{yaw}} + \lambda_{\text{map}}\mathcal{L}_{\text{map}} + \lambda_{\text{coll}}\mathcal{L}_{\text{coll}},$$

with loss weights λ_{yaw}, λ_{map}, and λ_{coll}. SHARP supervises the marginal predictions of its backbone alongside the scene-level joint global consistency module, and we retain this combination as $\mathcal{L} = 1.1\,\mathcal{L}_{\text{marg}} + 0.9\,\mathcal{L}_{\text{scene}}$.

2.3. Post-hoc Collision Disentanglement

The submission assigns the 32 trajectories of each agent into 32 distinct rollout slots. The kinematic term reshapes the rollout axis into a per-agent empirical distribution, and the coverage term is computed individually over each agent's rollouts. Consequently, both terms are invariant to permutations of the slot assignments when applied independently per agent. In contrast, the collision term is evaluated within a specific slot: an agent's rollout counts as colliding if its footprint overlaps with any other agent *in the same slot* at any time step. To reduce these collisions, we reorder the individual agent predictions across the rollout slots to minimize the total number of colliding agent-rollouts. Because this reordering does not alter any actual predicted trajectories, the kinematic, coverage, and valid-region components remain constant.

3. Experiments

Dataset. HetroD [4] provides 5,087 training, 955 validation, and 955 test scenarios recorded from a drone at 10 Hz. Each scenario spans 91 time steps, of which 11 are observed and 80 form the 8 s prediction horizon, and contains roughly 50 scored agents of the types vehicle, two-wheeler, and pedestrian. A submission must cover all of them, while the challenge metric is computed on four selected agents per scenario on average. We fine-tune our model on the HetroD training split and report results computed with the official toolkit on the validation set.

Implementation Details. Our implementation is based on the public code and model release of SHARP[1]. All models are trained on a single GPU with 96 GB of VRAM, using a batch size of two scenes, 16 scored agents per scene, and a learning rate of 10^{-4}. Fine-tuning starts from the pretrained SHARP Argoverse 2 [8] multi-agent checkpoint and follows the stages of Tab. 1. Initially, we add the individual refinements one after another and fine-tune for three epochs per stage. For a clean recipe, we train a single run of 20 epochs with all additions enabled for the final submission. The weights of the map compliance and collision terms are set such that each contributes 10% of the gradient norm of the regression term, measured on real training batches, which yields $\lambda_{\text{map}} = 0.82$ and $\lambda_{\text{coll}} = 23.6$. The heading loss weight is chosen empirically as $\lambda_{\text{yaw}} = 0.2$.

[1] https://github.com/a-pru/sharp

3.1. Model Adaptation Results

Table 1 reports our experimental results. As a simple baseline, we extrapolate every agent from its last valid observation at constant velocity, perturbing speed and yaw rate across the 32 rollouts so that the coverage term is not identically zero. Second, we report results using the pretrained SHARP AV2 multi-agent checkpoint without fine-tuning, applying the mode expansion of Sec. 2.1 and recovering the heading via finite differences along the predicted path. This zero-shot model improves the kinematic component over the constant-velocity baseline, but loses on cross-type consistency, as it is primarily trained on car trajectories.

Model Finetuning. Fine-tuning with the per-time-step masked objective establishes our first HetroD baseline. Replacing direct heading regression with integrated yaw rate provides our largest training gain, improving performance consistently across all five recording locations. Spatially neighborhood-based agent sampling during training further boosts the score, driven almost entirely by the yaw acceleration component (0.5540 to 0.6370). Supervising map compliance across all 32 modes primarily benefits the kinematic component rather than the valid-region metric itself. This indicates that the map effectively acts as a motion prior for the 31 non-winning modes, constraining their behavior without degrading acceleration realism. While collision supervision against surrounding traffic maintains the overall score, it further enhances kinematics and improves post-disentanglement collision metrics. Finally, balancing agent types during training targets the weakest submetric (pedestrian angular speed), yielding significant gains increasing from 0.3365 to 0.3979.

Final Submission. Fine-tuning the AV2 pretrained model for 20 epochs with all components enabled achieves a score of 0.7268 after post-hoc disentanglement, marginally worse than the initial staged schedule (0.7285). While the staged schedule maintains an advantage in kinematic realism (0.6849 vs. 0.6680), the single-stage model achieves stronger results on valid-region compliance (0.9756 vs. 0.9681), collision avoidance, and cross-type consistency. Favoring overall safety alongside a clean, easily reproducible recipe, we select the single-stage model for our final submission.

3.2. Collision Disentanglement

Table 2 shows the results of our post-hoc reordering of modes across rollout slots. The ratio of colliding agent-rollouts reduces from 32.10% to 17.15%, raising the collision score from 0.6943 to 0.8248 and the overall score by 0.0234. Kinematic realism and coverage remain constant, as they are invariant to the reordering. On the test split, where the procedure operates on the input scenarios instead of the ground truth, the ratio reduces from 28.29% to 16.64%, closely matching the validation result.

	Kinematic ↑	Safety ↑	**Score** ↑
Constant Velocity Baseline	0.2634	0.7566	0.5418
SHARP (AV2 Multi-Agent Zero-Shot)	0.3803	0.7806	0.5668
+ HetroD fine-tuning	0.5917	0.8247	0.6716
+ Yaw-rate Heading	0.6372	0.8259	0.6906
+ Spatial Agent Sampling	0.6527	0.8312	0.6966
+ All-mode map Compliance	0.6668	0.8326	0.7021
+ Surrounding-Traffic Collisions	0.6718	0.8317	0.7021
+ Type-Mixed Agent Sampling	0.6849	0.8237	0.7047
+ Collision Disentanglement	0.6849	0.8897	**0.7285**
Single-Stage Fine-Tuning (Ours)	0.6680	**0.9002**	0.7268

Table 1. **Adaptation ablation** on the full validation split of 955 scenes, evaluated using the official toolkit. Each row adds a single modification to the row above it. We report the kinematic realism term, a safety score combining both collision and valid-region metrics, and the final score, which also includes the cross-type consistency and coverage terms. The collision disentanglement module requires no training and runs purely as a post-hoc refinement. Our final submission trains a single model initialized from the pretrained AV2 multi-agent checkpoint with all enhancements enabled.

Metric Component	Direct	With Collision Disentanglement
Kinematic	0.6680	0.6680
Coverage	0.0941	0.0941
Valid Region	0.9756	0.9756
Cross-Type	0.8223	0.8229
Collision (Vehicles)	0.7252	**0.8694**
Collision (Two-Wheelers)	0.6550	**0.8168**
Collision (Pedestrians)	0.6911	**0.7877**
Colliding Agent-Rollouts	32.10%	**17.15%**
Overall Score	0.7034	**0.7268**

Table 2. **Effect of collision disentanglement** on the full validation split, evaluated before and after reassigning each agent's modes across the 32 rollout slots. An agent-rollout counts as colliding if its footprint overlaps with any other agent at any of the 80 predicted steps. Kinematic realism, coverage, and valid-region compliance remain constant, confirming that the individual trajectories are invariant and only their assignment across rollout slots changes.

4. Conclusion

We adapted SHARP [7], a streaming motion forecasting model, to the heterogeneous drone-recorded traffic of the HetroD Challenge [4]. Fig. 1 shows four scenes with the rollouts of our final model. We expanded the output dimension, added an integrated yaw-rate heading head, and updated both the loss formulation and the sampling of the scored agents. Together with the post-hoc collision disentanglement, our submission improves the challenge score from 0.57 zero-shot to 0.73 on the full validation split.

Acknowledgments. This work was partially funded by Addsafety (923936), a COMET Project funded by BMIMI, BMWET and the co-financing federal province of Styria. The COMET programme is managed by the Austrian Research Promotion Agency (FFG). Research supported by the NVIDIA Academic Grant Program using a NVIDIA RTX PRO 6000 Blackwell Workstation Edition.

References

[1] Hidde JH Boekema, Bruno KW Martens, Julian FP Kooij, and Dariu M Gavrila. Multi-Class Trajectory Prediction in Urban Traffic Using the View-of-Delft Prediction Dataset. *RAL*, 2024. 1

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A Multimodal Dataset for Autonomous Driving. In *CVPR*, 2020. 1

[3] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3D Tracking and Forecasting with Rich Maps. In *CVPR*, 2019. 1

[4] Yu-Hsiang Chen, Wei-Jer Chang, Christian Kotulla, Thomas Keutgens, Steffen Runde, Tobias Moers, Christoph Klas, Wei Zhan, Masayoshi Tomizuka, and Yi-Ting Chen. HetroD: A High-Fidelity Drone Dataset and Benchmark for Autonomous Driving in Heterogeneous Traffic. In *ICRA*, 2026. 1, 2, 3, 4

[5] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large Scale Interactive Motion Forecasting for Autonomous Driving: The Waymo Open Motion Dataset. In *ICCV*, 2021. 1

[6] Alexander Prutsch and Horst Possegger. Learning to Predict Mixed-Traffic Trajectories in Urban Scenarios from Little Training Data with Refined Environment Modeling. In *IEEE IV*, 2025. 1

[7] Alexander Prutsch, Christian Fruhwirth-Reisinger, David Schinagl, and Horst Possegger. SHARP: Short-Window Streaming for Accurate and Robust Prediction in Motion Forecasting. In *CVPR*, 2026. 1, 4

[8] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next Generation Datasets for Self-driving Perception and Forecasting. In *NeurIPS Datasets and Benchmarks*, 2021. 1, 3

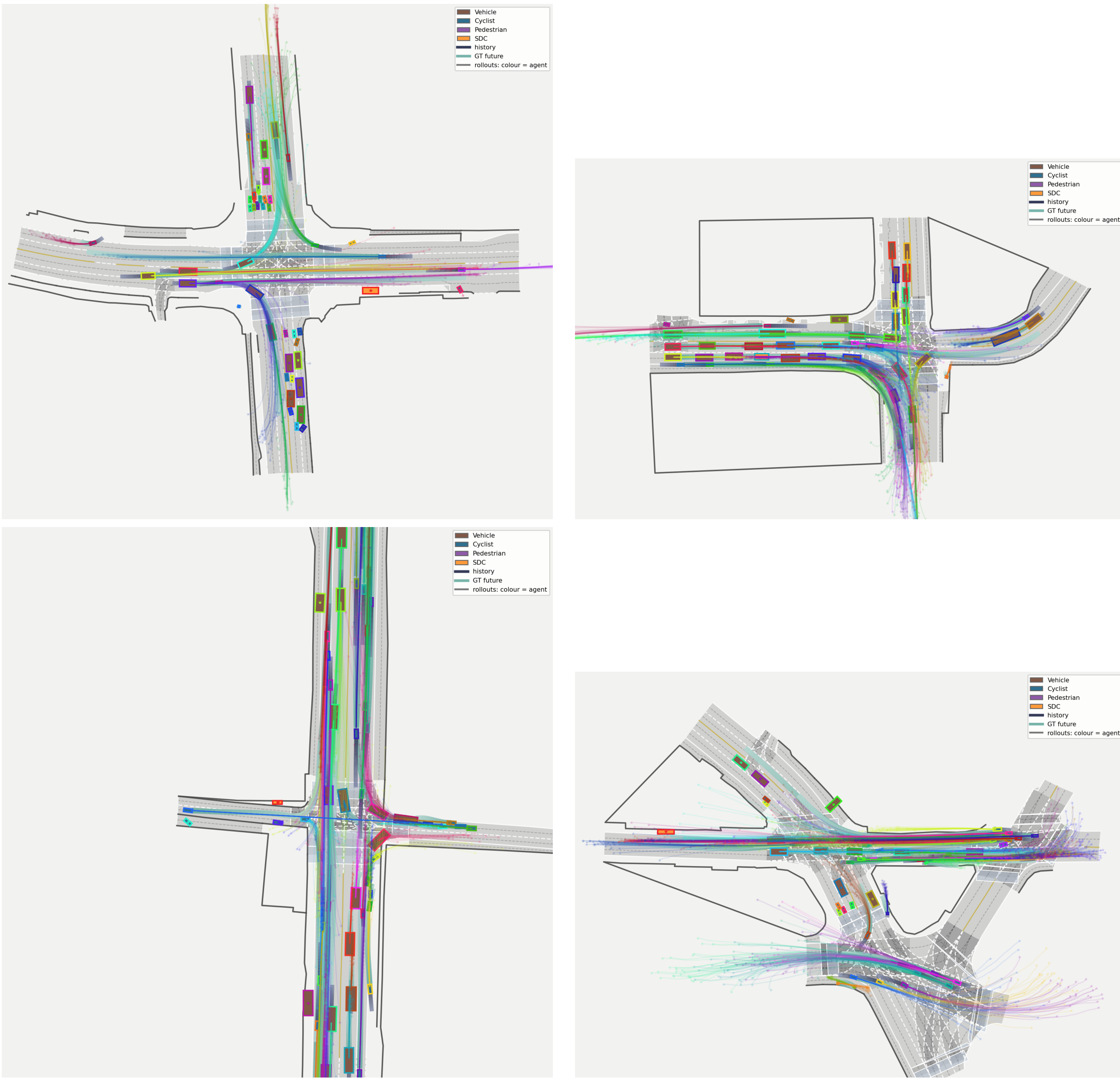

Figure 1. Rollouts of our final model on four validation scenes from different recording locations, simulated for 8 s. All 32 rollouts of every scored agent are drawn in the color of that agent, with a star marking the endpoint. Vehicles, two-wheelers, and pedestrians share the same intersections, and most agents barely move over the horizon while a few traverse the full scene. Thick bundles indicate agreement between the 32 hypotheses, while wide fans show agents whose maneuver is still ambiguous after the 11 observed steps.