HetroD-C1 Mix512-Safe: Multi-Scale SMART/CAT-K Rollout Selection for the HetroD Challenge

HetroD-C1 Team
Independent Research Team
HetroD-C1

Abstract. We describe the HetroD-C1 Mix512-Safe submission to the HetroD Challenge v1.0. Our final branch is a validation-selected SMART/CAT-K-style tokenized traffic simulation system adapted to heterogeneous traffic. It combines behavior cloning, closed-loop supervised fine-tuning, multi-temperature candidate generation, and a safety-first selector over 512 candidate rollouts. The final public full-validation score is 0.762271637492. We report the selected branch exactly as submitted and explicitly distinguish it from RFT/recovery variants that improved coverage diagnostics but did not improve the full-validation total.

Team and method. Our team name is HetroD-C1 and our method name is Mix512-Safe. The submitted method is a competition-oriented closed-loop rollout generation system for the HetroD Challenge v1.0. The final submitted branch is the validation-selected `mix512_k12t125_k24t18_k48t22 + safe` configuration. It uses a SMART/CAT-K-style trajectory-token backbone, HetroD-specific data adaptation, closed-loop fine-tuning, multi-temperature candidate generation, and a conservative safety-first selector that returns exactly 32 rollouts per test scenario. The goal of the system is not only to predict likely trajectories, but to generate a set of physically valid, interaction-aware, and sufficiently diverse closed-loop futures that matches the official evaluation protocol.

Team members and affiliations. The submission was prepared by the HetroD-C1 Team, Independent Research Team. No additional institutional affiliation is claimed for this submission.

Method overview. The final pipeline follows a three-stage design. First, a compact SMART/CAT-K trajectory-token model is adapted to HetroD and trained with behavior cloning on the official training split. SMART motivates the use of discrete trajectory tokens and next-token prediction for scalable multi-agent motion generation [2], while CAT-K motivates closed-loop supervised fine-tuning for reducing covariate shift without requiring reinforcement learning [3]. TrajTok further motivates combining rule-based coverage with data-driven filtering when constructing trajectory-token vocabularies [4]. Second, the best behavior-cloning checkpoint is refined by closed-loop supervised fine-tuning so that autoregressive rollout errors do not accumulate as aggressively over the 80-step future horizon. Third, inference generates a large pool of candidates and then performs metric-aware test-time selection. For the submitted branch, we merge four candidate banks, denoted `k12t125`, `k24t18`, `k48t22`, and the base sampling bank, giving up to 512 candidate rollouts per scenario before selection. The final selector applies hard quality gates for finite states, speed continuity, yaw smoothness, collision risk, and scene validity, then greedily chooses 32 rollouts that preserve coverage only among candidates that satisfy the safety and kinematic filters. This order was selected because the official metric strongly penalizes unsafe or implausible samples, and validation showed that unconditional diversity improves coverage but can reduce the final score.

Model architecture. The backbone is a SMART/CAT-K-style encoder-decoder for vectorized driving scenes. Map polylines and agent histories are embedded into a shared latent space and processed by stacked attention blocks. Agent-agent and agent-map interactions are represented by relative geometric features, and future motion is decoded in a trajectory-token space instead of by direct independent coordinate regression. This design follows the tokenized traffic-modeling direction used by SMART and CAT-K [2, 3]. The HetroD adaptation adds type-aware embeddings and type-conditioned decoding so that vehicles, two-wheelers, and pedestrians can retain different motion priors. This is important for HetroD because the benchmark is dominated by vulnerable road users and contains heterogeneous behaviors such as lane splitting, hook turns, informal right-of-way negotiation, and pedestrian hesitation [1]. The rollout head predicts future token sequences, which are decoded into global (x, y, z, yaw) states for every required agent.

Training procedure. Training starts from the official HetroD train split and uses the public validation split only for model selection and hyperparameter selection. The first training phase is behavior cloning with a weighted sampler. The sampler increases the frequency of scenes with cross-type pairs, two-wheeler density, pedestrian transition behavior, and high time-to-collision risk, while maintaining location balance to match the macro-averaged nature of the challenge. The second phase is closed-loop supervised fine-tuning from the best behavior-cloning checkpoint. During this phase, the model is trained closer to its rollout-time distribution so that small early errors do not create large final-horizon drift. We also tested metric-oriented RFT/DPO-style variants and a short supervised recovery phase, inspired by recent post-training work for tokenized traffic simulation [5, 6]. These variants increased candidate-set coverage in some diagnostics, but the selected full-validation score was lower than the Mix512-Safe branch because safety and cross-type interaction metrics degraded. Therefore, the final submission intentionally uses the validation-best Mix512-Safe branch rather than the lower-scoring RFT recovery branch.

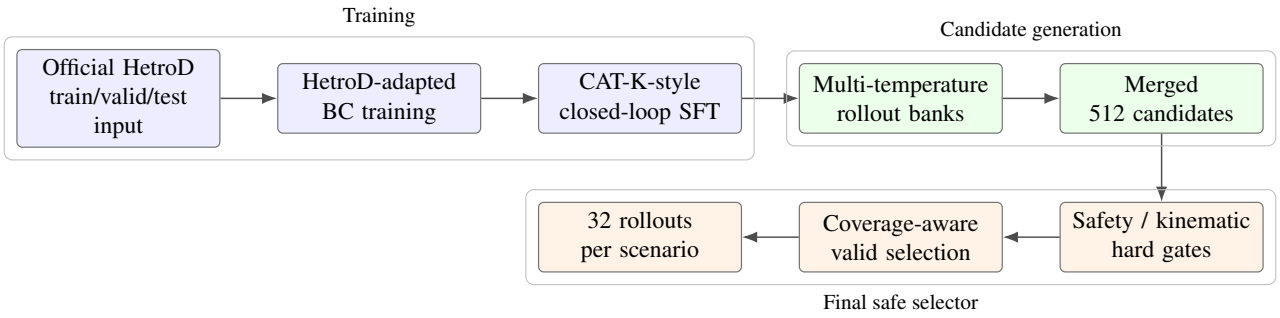

Figure 1: Final Mix512-Safe pipeline. The submitted branch uses the validation-best closed-loop checkpoint, merges four rollout banks into a 512-candidate pool, and selects exactly 32 valid rollouts after safety and kinematic gating.

Data and pretrained models used. The method uses the official HetroD Challenge v1.0 dataset for training, validation, and test input generation [1, 9]. Test-set ground truth is hidden and was not used. The system uses external public SMART/CAT-K code and pretrained trajectory-token modeling resources as initialization and implementation references [7, 8]. No hidden annotations, manual test labels, private map annotations, or external HetroD labels are used. The final form answer for external data/pretrained resources should therefore be **Yes**, because the method uses external public pretrained/model resources in addition to the official HetroD data.

Inference and rollout generation. At inference time, the model produces multiple stochastic rollout banks with different candidate counts and sampling temperatures. The submitted branch uses the merged `mix512_k12t125_k24t18_k48t22` candidate pool. Each candidate stores future steps 11–90 in global coordinates as $(x_m, y_m, z_m, yaw_{rad})$. The final safe selector returns exactly 32 rollouts per scenario and preserves the official `metadata.required_agent_ids` order. The selector first removes candidates with non-finite values, severe kinematic discontinuity, invalid yaw behavior, obvious collision risk, or scene-bound violations. It then selects a diverse subset from the remaining candidates using trajectory occupancy and endpoint spread, but only after the quality gates have been applied. This safety-first ordering was the best full-validation tradeoff found in our experiments.

Validation results. The final selected full-validation configuration is `mix512_k12t125_k24t18_k48t22 + safe`. On the public validation split with the official evaluation toolkit, it obtained a total score of 0.762271637492. The corresponding component scores were: Kinematic 0.767008, Safety 0.913754, Cross-type 0.794271, and Coverage 0.196719. A shard-level diagnostic for the same family reached 0.777880180399, which motivated the final test rerank using the same safety-first setting. More exploratory source-aware and RFT-style selectors increased coverage on some subsets, but they did not improve the full-validation total score. The submitted test package therefore follows the strongest full-validation branch, not the highest-coverage diagnostic branch.

Submission format. The produced challenge submission contains one top-level folder of scenario-level pickle files, one file for every scenario listed in `manifests/test.txt`. Each pickle contains `agent_id` with the exact required agent IDs and `simulated_states` with shape $[32, num\_agents, 80, 4]$. We run the official public preflight script before upload to check file count, scenario IDs, required agent IDs, shapes, floating-point dtypes, and finite values.

References

[1] Y.-H. Chen, W.-J. Chang, C. Kotulla, T. Keutgens, S. Runde, T. Moers, C. Klas, W. Zhan, M. Tomizuka, and Y.-T. Chen. HetroD: A High-Fidelity Drone Dataset and Benchmark for Autonomous Driving in Heterogeneous Traffic. arXiv:2602.03447, 2026. 1, 2

[2] W. Wu, X. Feng, Z. Gao, and Y. Kan. SMART: Scalable Multi-agent Real-time Motion Generation via Next-token Prediction. arXiv:2405.15677, 2024. 1

[3] Z. Zhang, P. Karkus, M. Igl, W. Ding, Y. Chen, B. Ivanovic, and M. Pavone. Closed-Loop Supervised Fine-Tuning of Tokenized Traffic Models. arXiv:2412.05334, 2024. 1

[4] Z. Zhang, X. Jia, G. Chen, Q. Li, and J. Yan. TrajTok: Technical Report for 2025 Waymo Open Sim Agents Challenge. arXiv:2506.21618, 2025. 1

[5] M. Pei, S. Shi, and S. Shen. Advancing Multi-agent Traffic Simulation via R1-Style Reinforcement Fine-Tuning. arXiv:2509.23993, 2026. 1

[6] E. Ahmadi, H. Schofield, B. Khamidehi, F. Arasteh, J. Shan, L. Mou, D. Bai, and K. Rezaee. RLFTSim: Realistic and Controllable Multi-Agent Traffic Simulation via Reinforcement Learning Fine-Tuning. CVPR, 2026. 1

[7] SMART project. Official SMART implementation. https://github.com/rainmaker22/SMART. 2

[8] CAT-K project. Official CAT-K implementation. https://github.com/NVlabs/catk. 2

[9] HetroD Challenge organizers. HetroD Challenge v1.0 Evaluation Toolkit. https://github.com/bob020416/HetroD-Challenge-v1.0-Evaluation.

2