HeteroTLA: Type-Conditioned Latent Augmentation for Heterogeneous Joint Motion Forecasting

Yancy Team

Yingxin Wang, Jianing Liu, Yiwen Ren, Xinglin Xie, Qin Ma,

Fang Liu, Wenping Ma, Puhua Chen

Joint International Research Laboratory of Intelligent Perception and Computation

Abstract

We describe HETEROTLA, a heterogeneous joint motion forecasting system combining a four-model autoregressive ensemble, type-conditioned latent residual augmentation, and safety-gated candidate selection. Three project-specific HeteroSMART variants and a BehaviorGPT-style predictor generate 256 joint trajectory candidates. A conditional variational residual decoder attached to the BehaviorGPT branch supplies 64 additional candidates, with residuals enabled for two-wheelers and pedestrians. The evaluated configuration constructs a 32-mode reference set through a strict proxy-based selector, retains its first 24 modes, and fills up to eight positions from the augmented pool with deterministic fallback. On all 955 schema-1.3 public-validation scenes, the official `hetrod-0.8.0` evaluator reports 0.6617455786, compared with 0.6613856924 for the strict reference. The absolute gain is 0.0003598862. This comparison is reported on the complete public-validation split.

1 Introduction

Forecasting heterogeneous traffic requires modeling interactions among vehicles, two-wheelers, and pedestrians while representing multiple plausible futures. For vulnerable road users (VRUs), directional changes and lateral motion can be poorly represented by a small set of dominant modes. In joint simulation, increasing diversity must also be balanced against kinematic plausibility and interactions between agents.

Trajectory-set approaches such as CoverNet [1] motivate explicit candidate generation and selection. Trajectron++ [2] addresses heterogeneous dynamics, and LaneGCN [3] illustrates the value of structured map context. Scene Transformer [4] models multiple agents jointly. SMART [5] and BehaviorGPT [6] provide related autoregressive motion-generation formulations, while MotionDiffuser [7] studies joint generative prediction. These works provide context; their published architectures and benchmark results are not claimed as reproduced here.

Our system combines locally trained predictors with a conditional latent decoder and trajectory-level selection. We name the combined approach *Heterogeneous Type-conditioned Latent Augmentation* (HETEROTLA). Its practical objective is to expand a candidate pool while retaining a fixed subset of an established reference prediction. The report describes the implemented configuration and its measured public-validation result rather than asserting a new state-of-the-art architecture.

2 Method

Figure 1 shows the configuration associated with the reported validation score. Each mode is a *joint* rollout for all required agents; 32 modes therefore represent 32 alternative scenes rather than independent per-agent top-K forecasts.

2.1 Frozen Heterogeneous Predictors

The ensemble contains three project-specific HeteroSMART variants (spatial, autoregressive/reflection, and map/codebook) and one BehaviorGPT-style next-patch predictor. The four networks have independent parameters and scene encoders. The variant names describe local implementations, not official pretrained releases of the cited methods.

The BehaviorGPT-style branch uses hidden width 128, eight attention heads, two attention blocks, 16 patch modes, a 10-step prediction patch, and five-step replanning. Its context includes agent history, local map geometry, and neighboring agents, with at most 32 neighbors. Forward, lateral, and yaw controls are integrated to produce 80 future states at 10 Hz. Each branch contributes 64 candidates, yielding a 256-mode ensemble pool $\mathcal{C}_0$. Candidate generation samples model outputs and merges trajectories in a common coordinate frame.

The production path loads supervised prediction checkpoints; it contains no policy-gradient or diffusion checkpoint. Reflection, autoregressive decoding, and map/codebook modeling are branch-specific design choices. We do not assume that every training objective or augmentation was applied identically to all branches. Several frozen checkpoint configurations refer to an earlier public-data root, so the system should be described as schema-1.3 adaptation of existing predictors, rather than end-to-end schema-1.3 retraining.

2.2 Type-Conditioned Latent Augmentation

TLA is a conditional variational autoencoder (CVAE)-style residual module attached to frozen BehaviorGPT features. For batch size B, N agents, R replanning patches, and five control

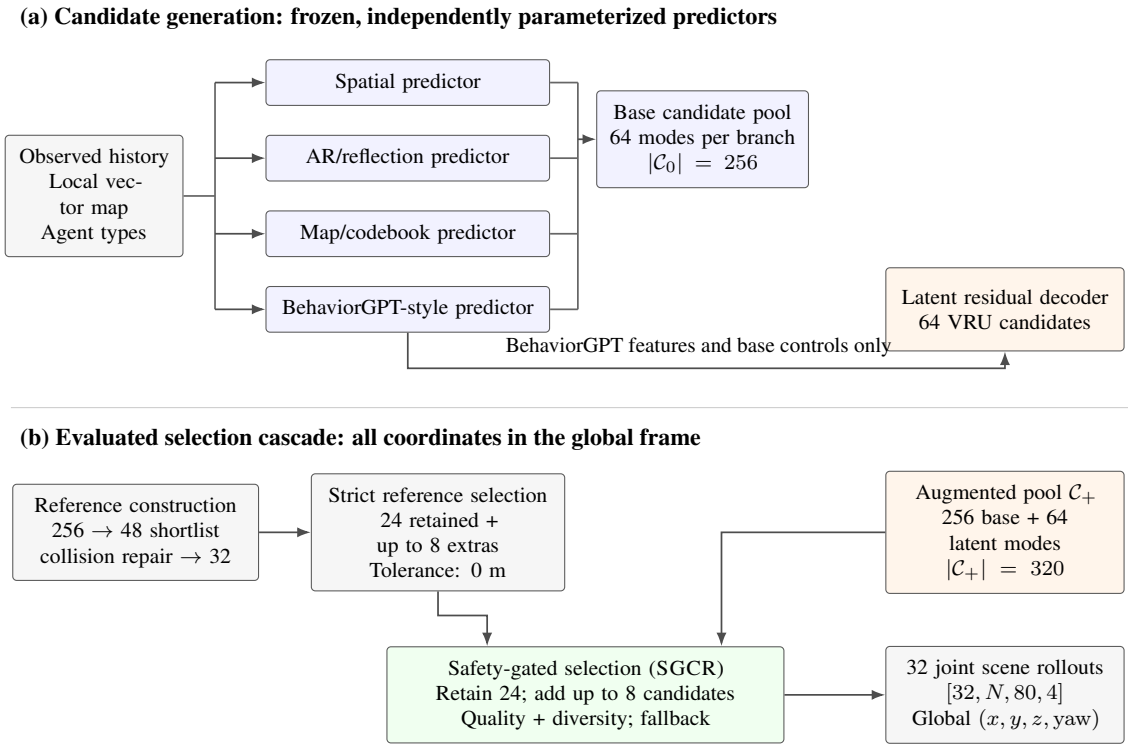

Figure 1: Configuration evaluated at 0.6617455786 on public validation. The models have separate encoders; latent augmentation is attached only to the BehaviorGPT-style branch. The selection cascade is drawn separately for readability.

steps per patch, its inputs are

$$h \in \mathbb{R}^{B\times N\times R\times 128}, \qquad u \in \mathbb{R}^{B\times N\times R\times 5\times 3}.$$

A context projection combines h with mean patch controls. Small multilayer perceptrons parameterize a conditional Gaussian prior, a target-conditioned posterior for training, and a bounded residual decoder. At inference,

$$u'_{i,r,t} = \text{clip}\big(u_{i,r,t} + \alpha_{\tau_i}\, g_\theta(h_{i,r}, u_{i,r}, z_r)_t\big). \tag{1}$$

The implemented sampler draws $z_r \sim \mathcal{N}(0, I_{24})$ independently per patch and shares it across agents within each rollout. Thus it introduces shared stochastic input, but does not enforce a single persistent scene intent over the entire horizon. It also bypasses the learned conditional prior at inference; this train–inference mismatch is a limitation.

Before type scaling, the decoder bounds each step's forward, lateral, and yaw residual by 0.90 m, 0.85 m, and 0.24 rad. We set $\alpha_{\text{VRU}} = 1.25$ and $\alpha_{\text{vehicle}} = 0$. Subsequent control clipping uses the backbone bounds of 3.0 m, 1.5 m, and 0.7 rad. Zero vehicle scaling means no direct vehicle residual is added; it is not a collision-free or unchanged-scene guarantee. The BehaviorGPT branch contributes 64 additional latent-decoded candidates, giving $|\mathcal{C}_+| = 320$.

2.3 Training

The latent module is trained with the BehaviorGPT backbone frozen on 5,087 schema-1.3 training scenes. Training uses target-conditioned posterior reconstruction, best-of-six conditional-prior sampling, Gaussian KL regularization, and an endpoint diversity margin:

$$\mathcal{L} = \mathcal{L}_{\text{var}} + 0.20\mathcal{L}_{\text{rec}} + 0.01\mathcal{L}_{\text{KL}} + 0.02\mathcal{L}_{\text{div}}. \tag{2}$$

The variety term minimizes patch ADE plus 0.5 times patch FDE over six sampled controls. Its agent weights are 2.5 for VRUs and 0.75 for vehicles, with an additional factor of 2 for selected-proxy agents. The diversity margin is 0.7 m. Future targets are used for supervised training and validation loss, but are not inputs to inference or candidate selection.

The run records five epochs, batch size 1, learning rate 3×10^{-4}, weight decay 10^{-4}, and gradient clipping at 1.0. The minimum logged validation loss is 0.0340216. This training loss is distinct from the official challenge score. Checkpoint selection uses public validation; the final comparison is consequently a development-set comparison.

2.4 Reference Construction and Safety-Gated Selection

The initial inference pipeline shortlists 48 of 256 base candidates using quality and diversity, applies collision repair, and selects

32 reference modes. For the evaluated validation configuration, a further selector constructs a strict reference set $\mathcal{B}$: it preserves 24 modes and selects up to eight from $\mathcal{C}_0$, using collision proxy tolerance 0 m.

The final SGCR stage retains the first 24 modes of $\mathcal{B}$ and considers $\mathcal{C}_+$ for eight additional positions. It scores each candidate c using

$$\begin{aligned} q(c) = & -14r_{\text{coll}}(c) - 0.30p_{\text{kin}}(c) \\ & - 0.08p_{\text{off}}(c) - 0.03p_{\text{trend}}(c). \end{aligned} \tag{3}$$

Here p_{off} approximates vehicle departure using distance to sampled map surface points, and p_{trend} measures first-step velocity mismatch with observed history. These are local ranking proxies, not the official scoring components. Admission requires

$$\begin{aligned} & r_{\text{coll}}(c) \le \text{mean}_{b\in\mathcal{B}}\, r_{\text{coll}}(b), \\ & p_{\text{kin}}(c) \le 1.03\,\text{mean}_{b\in\mathcal{B}}\, p_{\text{kin}}(b) + 10^{-6}. \end{aligned} \tag{4}$$

For this final stage, collision overlap tolerance is 0.1 m. Greedy selection combines scene-normalized quality with agent-wise embedding diversity weighted by 0.55. If fewer than eight candidates pass, unused reference modes fill the remaining positions. Final selection copies trajectory coordinates; the earlier reference construction includes collision repair. The eight extras may come from either base or latent candidates, and need not represent eight new behavioral modes.

2.5 Inference and Output

Inference uses the observed history, agent attributes, and map context. The published input contract contains 11 history steps; predicted states correspond to future steps 11–90. The final array is float32 with shape $[32, N, 80, 4]$, containing global (x, y, z, yaw) in the required agent order. Height is carried from the current observed state. The output has exactly 32 modes regardless of the number of candidates admitted by the selector.

3 Evaluation

3.1 Data and Protocol

The schema-1.3 public data contain 5,087 training scenes and 955 public-validation scenes. The extracted data passed the public-data validator. Results below use the official `hetrod-0.8.0` evaluator [8], with 955 matched validation scenarios, zero skips, and zero errors. The official aggregate includes kinematic realism, safety, cross-type interaction, and coverage-related scoring, with location-level macro aggregation.

3.2 Public-Validation Results

Table 1 gives the observed difference between the strict reference and the complete cascade. This comparison changes both the candidate pool and the selection stage. It does not isolate the causal effect of latent generation alone. No confidence interval or repeated-seed significance analysis is available, so the small improvement is reported as an observed development result.

Table 1: Full schema-1.3 public-validation comparison, official v0.8.0; both rows use 955 scenes.

Configuration	Official score
Strict reference selector	0.6613856924
HETEROTLA selection cascade	**0.6617455786**
Absolute difference	+0.0003598862

Table 2: Official components of the evaluated HETEROTLA configuration.

Component	Score
Kinematic realism	0.6940498086
Safety	0.7368789820
Cross-type interaction	0.7350038806
Coverage	0.2331993038

Coverage is numerically the lowest reported component, suggesting a useful diagnostic direction. Different component definitions prevent this observation alone from establishing the dominant cause of the aggregate score. Scores by location are 0.6984996214, 0.6211389373, 0.6510280447, 0.6759323851, and 0.6621289044 for loc0, loc1, loc3, loc4, and loc5, respectively. Per-location improvements cannot be inferred from these values without matching reference comparisons.

The official diagnostics still report non-zero new-collision rates: approximately 0.4786 for vehicles, 0.4596 for two-wheelers, and 0.3481 for pedestrians. Proxy gates therefore do not establish physical safety. Patch ADE/FDE used in training should not be presented as measured full-horizon validation ADE/FDE; those metrics are not reported here.

4 Limitations and Compliance

Evaluation scope. The reported comparison uses the complete schema-1.3 public-validation split and the official v0.8.0 evaluator. Both configurations use the same 955 scenes and the same metric implementation. Checkpoint and selector choices were fixed before the final evaluation record was produced, and all 955 evaluations completed without skips or errors.

Safety and distributional assumptions. Local collision and kinematic gates are heuristic proxies. Even zero overlap tolerance does not reproduce the official selected-anchor/full-context safety procedure. The final 0.1 m tolerance further differs from strict overlap accounting. The CVAE uses target-conditioned training and standard-normal, patchwise shared sampling at inference, which can limit calibration and temporal consistency. No formal dynamics, off-road, or collision-free guarantee is claimed.

Data, selection, and attribution. All training samples, annotations, and map data used by the reported pipeline come from the official HetroD public release. The latent module uses the schema-1.3 training split, and the motion codebook is constructed from the same source. The frozen checkpoints were

trained in-house; their initializations trace to earlier in-house runs on the official public release. No third-party pretrained weights or external annotations are used. SMART and BehaviorGPT are cited as related methods; the checkpoints are project implementations.

5 Conclusion

HETEROTLA combines a heterogeneous autoregressive ensemble, CVAE-style VRU residual generation, and a two-stage candidate selection cascade. The complete configuration scores 0.6617455786 on all 955 schema-1.3 public-validation scenes using the official v0.8.0 evaluator. The result reflects the measured behavior of the full candidate-generation and selection pipeline on that split.

References

[1] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff. CoverNet: Multimodal Behavior Prediction Using Trajectory Sets. In *CVPR*, 2020.

[2] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone. Trajectron++: Dynamically-Feasible Trajectory Forecasting With Heterogeneous Data. In *ECCV*, 2020.

[3] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun. Learning Lane Graph Representations for Motion Forecasting. In *ECCV*, 2020.

[4] J. Ngiam et al. Scene Transformer: A Unified Architecture for Predicting Multiple Agent Trajectories. In *ICLR*, 2022.

[5] W. Wu, X. Feng, Z. Gao, and Y. Kan. SMART: Scalable Multi-agent Real-time Motion Generation via Next-token Prediction. In *NeurIPS*, 2024.

[6] Z. Zhou, H. Hu, X. Chen, J. Wang, N. Guan, K. Wu, Y.-H. Li, Y.-K. Huang, and C. J. Xue. BehaviorGPT: Smart Agent Simulation for Autonomous Driving with Next-Patch Prediction. *arXiv:2405.17372*, 2024.

[7] C. M. Jiang, A. Cornman, C. Park, B. Sapp, Y. Zhou, and D. Anguelov. MotionDiffuser: Controllable Multi-Agent Motion Prediction Using Diffusion. In *CVPR*, 2023.

[8] HetroD Challenge Evaluation. Public competition evaluator, v0.8.0. Official evaluator repository, release v0.8.0.