INTERACTION-CONDITIONED NAVIGATION FOR END-TO-END AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reactive driving requires temporary navigation changes in response to other road users, but trajectory imitation does not explicitly supervise the navigation condition connecting an interaction to its execution. CondAD localizes candidate actors, relates them to the ego route, and verifies their identity on detection-guided image crops. The actor–route interaction becomes a navigation condition, a maneuver command together with a spatial target, and a persistent maneuver state keeps that condition across planning steps. On Bench2Drive, Give Way success rises from 40.00% to 90.00% (9/10). Progressive ablations on that set support converting trigger information into a command–target condition and retaining the maneuver state during execution.

1 INTRODUCTION

End-to-end autonomous driving has evolved from direct sensor-to-control prediction toward unified architectures that expose structured scene representations to planning. Modern systems jointly model surrounding actors, road geometry, ego state, motion, and route guidance through task queries or sparse scene representations (Hu et al., 2023; Jiang et al., 2023; Jia et al., 2025; Tang et al., 2025). These architectures allow a planning query to access rich information about other road users. However, access to interaction evidence is not equivalent to supervision of its role in a driving decision. The dominant trajectory target specifies where the expert moved, but does not uniquely identify which road user, or which relation to that road user, made the maneuver necessary.

Consider an ego vehicle traveling along an unobstructed lane while an emergency vehicle approaches from behind. The route still points forward. Yielding requires a temporary navigation change: create space, hold that space while the emergency vehicle passes, and return when the interaction permits. The recorded trajectory states where the expert moved. It does not label the interaction that induced the change (Figure 1).

Reactive driving asks for a temporary navigation change. The nominal route can stay in place while another road user requires the ego vehicle to deviate and hold that deviation. Trajectory imitation supervises the path that was driven. It does not supervise the navigation condition that connects the interaction to that path.

CondAD exposes the condition as a maneuver command and a spatial target, grounded in the actor and the lane relation, and it retains the maneuver state while the maneuver applies. The evaluation then looks at closed-loop behavior, and at how success changes as the trigger is converted into a command–target condition and the maneuver state is kept. Figure 1 illustrates the missing label. It does not measure the cause of a trajectory.

2 RELATED WORK

2.1 REPRESENTATIONS FOR END-TO-END PLANNING

End-to-end driving no longer refers only to a direct mapping from sensor observations to control. Recent systems expose increasingly structured scene representations to the planner. TransFuser fuses sensor features for imitation-based driving, while UniAD connects detection, tracking, prediction, and planning through a unified query-based architecture (Chitta et al., 2023; Hu et al., 2023).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

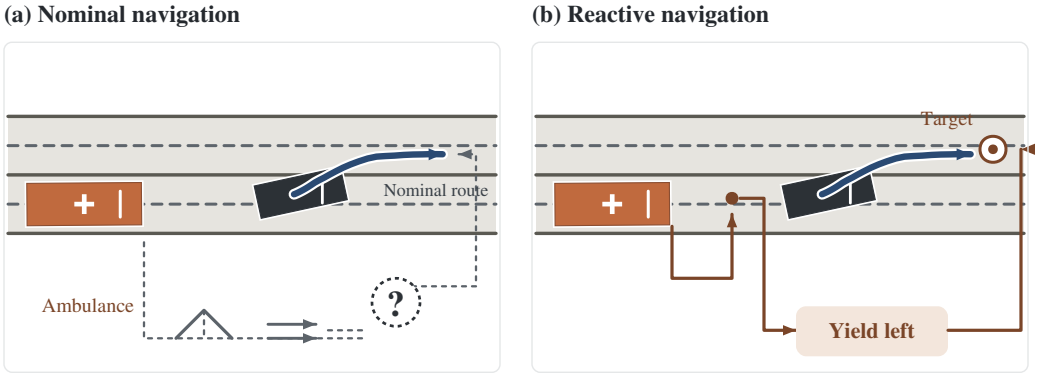

Figure 1: **Trajectory supervision does not label the interaction.** Both panels show the same recorded Bench2Drive scene, where an emergency vehicle approaches from behind while the nominal route remains forward. (a) The blue curve is the recorded expert motion. Trajectory supervision does not explicitly label the interaction that induces the temporary navigation change. (b) The drawing associates that actor with a temporary maneuver command and spatial target, illustrating a navigation condition before trajectory execution. The figure illustrates the distinction between trajectory supervision and interaction-conditioned navigation. Solid lines bound the two same-direction lanes. The gray dashes mark their centers; the labeled dash is the map-derived nominal reference, and the blue curve is three seconds of recorded expert motion. Command and target annotations illustrate the condition; ordinary traffic is omitted for clarity.

VAD and SparseDrive represent surrounding agents and road geometry with compact vectors or sparse instances (Jiang et al., 2023; Sun et al., 2025). DriveTransformer further organizes agent, map, and planning queries in a scalable unified transformer, whereas HiP-AD retrieves multi-view image evidence according to hierarchical planning geometry (Jia et al., 2025; Tang et al., 2025).

These representations make actors, maps, and route context available to the planner. An actor query can still enter planning without being identified as the reason the ego vehicle leaves its nominal behavior.

2.2 Interaction Modeling and Decision Attribution

How surrounding road users influence ego planning has been studied from several directions. GameFormer models iterative responses among agents through game-theoretic interaction, and PPAD repeatedly exchanges information between prediction and planning (Huang et al., 2023; Chen et al., 2025). UniAD and VAD likewise allow agent, motion, map, and planning representations to interact within a shared driving model (Hu et al., 2023; Jiang et al., 2023). Think Twice Before Driving retrieves safety-relevant scene evidence with an initial trajectory, while Causal-Planner uses episodic memory gating to disentangle interaction information (Jia et al., 2023b; Yuan et al., 2025). These methods improve how traffic participants are represented or exchanged with the planner. They do not necessarily make explicit which actor–route relation should be held responsible for a particular change in navigation intent.

A related concern is causal confusion in imitation-based driving. DriveAdapter studies the coupling between perception and planning when transferring privileged driving knowledge, and perception-guided causal supervision seeks to reduce reliance on shortcut correlations (Jia et al., 2023a; Huang et al., 2025). Dataset analyses similarly show that driving policies can exploit biases that reproduce expert motion without recovering the intended decision process (Zimmerlin et al., 2024). Temporal planning methods address a different but complementary issue: BridgeAD preserves historical prediction and planning context, MomAD transfers planning momentum, and CogDriver promotes coherent multi-step decisions (Zhang et al., 2025; Song et al., 2025; Liu et al., 2026).

The question here is narrower. It is whether a specific road user's relation to the current route is tied to a temporary maneuver condition that stays active through execution.

2.3 Driving Intent as a Planning Condition

A second line of work studies how route, goal, or instruction representations shape trajectory generation. Navigation-guided sparse scene representation prioritizes scene elements according to route information (Li & Cui, 2025). GoalFlow generates scene-consistent goal points and conditions multimodal trajectory generation on the selected goal (Xing et al., 2025). LEAD shows that a single target point can under-specify expert intent, particularly for maneuvers that unfold over multiple steps (Nguyen et al., 2026). Language-conditioned driving provides a richer interface for expressing intent. SimLingo aligns visual understanding with driving actions, while ORION uses vision–language instructions to guide action generation (Renz et al., 2025; Fu et al., 2025). Such methods show that maneuver-level information can be introduced before continuous trajectory decoding, rather than being recovered only from motion supervision.

The condition studied in this work differs in both its source and its lifetime. It is not a static destination, a map-derived route command, or a free-form instruction. It is activated by a localized actor–route interaction, represented jointly by a maneuver command and a spatial target, and maintained until the temporary maneuver is completed.

3 Method

3.1 Overview

CondAD separates the evidence for a reactive maneuver from the motion that executes it. Figure 2 summarizes the design: panel (a) locates a decision-relevant actor and verifies its identity; panel (b) turns the resulting evidence into a navigation instruction and trains the planner to follow it. The instruction specifies a maneuver before its continuous trajectory is generated.

We build on the query-based architecture of HiP-AD (Tang et al., 2025). Multi-view images are encoded into image features. Detection, map, and planning queries keep their existing route through the backbone. The two decoder blocks in the top panel show processing before and after conditioning within the driving backbone, rather than two standalone perception and planning systems.

The added path is the purple query stream. A yield query carries actor-grounded evidence, and an instruction query carries the active navigation condition. The instruction query conditions the planning query and is carried to the next planning step. Actor grounding is followed by activation, then by a command and a target, and then by a persistent instruction that conditions path and speed.

3.2 Grounding the Yield Trigger

Route-grounded actor selection. The upper row of Figure 2(a) starts with the detection head, which provides a box $b_{t,i}$, coarse category $\ell_{t,i}$, and associated query $\mathbf{q}_{t,i}^{\mathrm{det}}$ for each candidate i. Selection considers candidates across the scene rather than searching only an ego-front region. The model's own map and detection associate each actor with the lane-relative quantities d_s, d_d, and d_v, and select the relevant actor. Those quantities describe where an actor is and whether it is approaching. The lane-relation view and short track sequence illustrate these quantities. A nearby vehicle on an unrelated lane need not carry the same significance as one approaching along the decision lane.

The pair map preserves the correspondence between a candidate and the lane relation used to select it. Ground-truth scene annotations construct the training pairs, as indicated by the dashed *GT pairs* branch; this is not an additional learned matching objective. At deployment, association uses detected actors and the lane geometry available to the system, rather than GT actor identities. The selected box, query, and coarse category remain attached to the same candidate when passed to the next stage. Selection establishes relevance, not emergency-vehicle identity.

Detection-guided RGB verification. The lower row of Figure 2(a) addresses identity separately. A correctly localized van, for example, is not yet identified as an ambulance. Rather than asking the coarse detection category to resolve this distinction, we return to the original camera image. The selected box is projected into a valid camera view, cropped, and resized to the classifier input. The coarse category routes that crop to one binary verifier, which returns an identity probability.

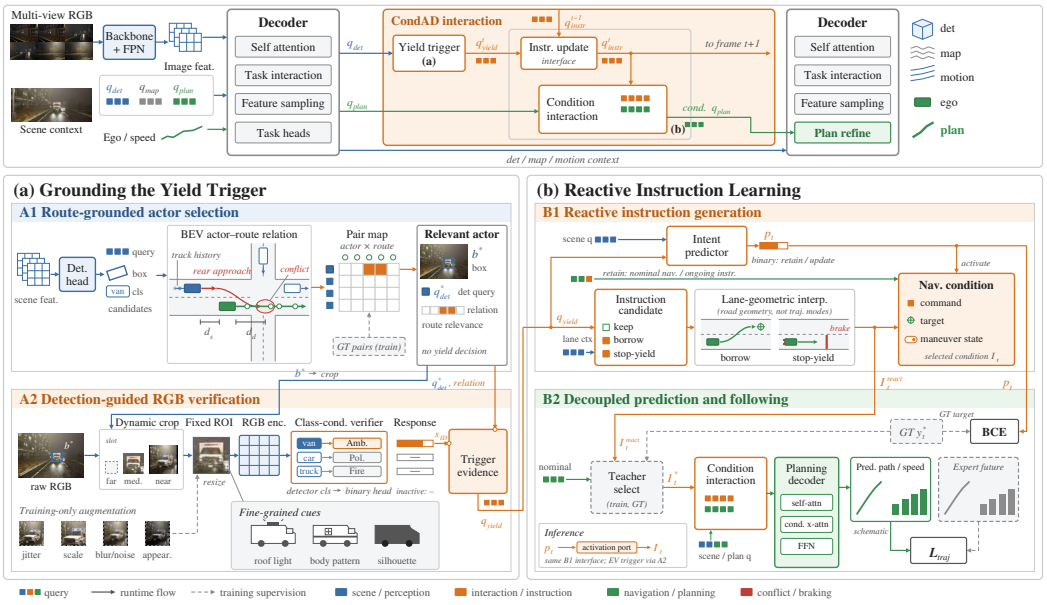

Figure 2: **CondAD: actor-grounded instructions for reactive driving.** The top panel adds a temporally carried instruction representation to the HiP-AD query pipeline. (a) Global detection and lane-relative pairing select actors for native-RGB, coarse-class-routed identity verification. (b) Reactive instructions are interpreted using lane geometry. GT labels supervise activation prediction and separately select the teacher condition for trajectory imitation. Solid arrows show forward dataflow; dashed arrows show training-only inputs or supervision. Photographs are recorded examples; matrices, bars, and vector trajectories are schematic.

The illustrated routing associates van, car, and truck candidates with ambulance, police-vehicle, and fire-engine verification, respectively; ordinary vehicles within each coarse category provide the corresponding negative examples.

The far/near crops and fixed ROI illustrate scale normalization, not super-resolution. The classifier receives the pixels available in the original image before detection-feature downsampling; resizing does not recover detail absent from those pixels. During training, crop jitter, scale perturbation, and blur augment these inputs to represent box misalignment and appearance degradation. This augmentation branch is removed at inference. The verified identity evidence stays associated with the selected detection and its route relation. The verifier score stays attached to that actor and does not by itself trigger the maneuver. The resulting representation, q_{yield}, is passed to instruction generation. Neither identity nor proximity alone is treated as a complete maneuver instruction.

3.3 Reactive Instruction Generation

The upper row of Figure 2(b) separates three quantities: whether navigation should change, the intended maneuver, and its geometric realization. The activation output is binary. That bit decides whether navigation changes. It is not a three-way classifier over *keep*, *borrow*, and *stop-yield*, and it does not regress a waypoint or name a target lane. The maneuver command is then read from the interaction and the lane relation, and the spatial target is read from the current lane geometry. The *keep*, *borrow*, and *stop-yield* entries denote instruction meanings, not trajectory modes predicted by the binary head.

The instantiated instruction is a navigation condition with three parts: a command, a geometric reference, and a maneuver state. Lane geometry and the nominal route interpret the current instruction, and the maneuver state carries the previous condition forward. This step sets the condition. It does not regress a waypoint or maintain a learned recurrent state. The instruction query is the representation of that command, geometric target, and maneuver state on which the planning query is conditioned.

The two lane diagrams show why this intermediate instruction is useful. For *borrow*, an adjacent driving lane is selected, its centerline provides the navigation target, and the navigation command specifies the corresponding lane change. The selected lane is retained during execution, while its forward target is recomputed from the current geometry. For *stop-yield*, the reference is the intersection stop line. Its longitudinal distance determines an explicit braking cue at the control interface; yielding is not represented by inserting a zero-speed trajectory point. Without an active reactive maneuver, the interface retains nominal navigation. Thus the geometric reference realizes an instruction; it does not define the instruction's meaning on its own.

The temporal connection in the top panel carries the instruction and its associated state. It retains the ongoing maneuver, rather than freezing an old target coordinate as the ego vehicle moves. The carried state retains the maneuver and its geometric reference. The target is recomputed from the current geometry at each step, rather than held as a frozen coordinate. When the maneuver no longer applies, the condition is released and nominal navigation remains. The instruction query conditions the planning query. The planner then produces path and speed, written $\widehat{\mathcal{W}}_t$, from the image features, the detection and map queries, and this conditioned planning query. The instruction does not replace those outputs. Scene context remains available to the planner when it executes the maneuver.

3.4 Decoupled Prediction and Following

The lower row of Figure 2(b) uses a demonstration for two different tasks: predicting whether to activate an instruction and executing a supplied instruction. Offline annotations associate scene actors and lane relations with expert behavior to provide an activation label $y_t^* \in \{0, 1\}$ and the expert motion target $\mathcal{W}_t^*$. Future information used to construct these training targets is not an input to the activation predictor.

Teacher-conditioned execution. The instruction interface supplies nominal and reactive candidates, $\mathcal{I}_t^{\text{nom}}$ and $\mathcal{I}_t^{\text{react}}$, in the same representation. For each training sample, the GT label selects

$$\mathcal{I}_t^* = \begin{cases} \mathcal{I}_t^{\text{nom}}, & y_t^* = 0, \\ \mathcal{I}_t^{\text{react}}, & y_t^* = 1. \end{cases} \tag{1}$$

The selected condition is encoded as the instruction query and conditions the planning query. Selection is discrete: commands and target coordinates are not averaged using a predicted probability. The reactive candidate is instantiated by the lane interpreter, rather than by copying the expert trajectory into the navigation input. Expert motion supervises the motion produced under this condition.

Intent prediction. The parallel branch reads features that are observable before teacher conditioning and outputs an activation probability p_t. For the borrowing predictor, these features include side-view and left-lane detections. The logit is computed before the navigation command or target is rewritten. Neither y_t^*, the selected teacher condition, nor expert future motion is an input to this head. The two supervision terms are

$$\mathcal{L}_{\text{intent}} = \text{BCE}(p_t, y_t^*), \qquad \mathcal{L}_{\text{exec}} = \ell_{\text{plan}}\left(\widehat{\mathcal{W}}_t, \mathcal{W}_t^*\right). \tag{2}$$

ℓ_{plan} denotes the planner's trajectory-supervision objective, evaluated on the applicable expert targets and valid future samples. The activation label supervises maneuver selection, whereas expert motion supervises its continuous realization.

The GT label serves as the intent target and the execution selector. Since teacher selection does not depend on p_t, the trajectory loss has no path through this selector to the activation logit. Shared encoder parameters may still receive planning gradients; separate supervision paths do not imply independent optimization. Teacher-selected navigation is excluded from the predictor's input.

Deployment. No GT label is used during driving. The borrowing predictor selects conditions from its activation score, whereas emergency-vehicle activation uses the crop verifiers and lane relation in panel (a). These distinct predictors share the instruction interface. The selected instruction is interpreted in current lane geometry and maintained across planning steps. Ground-truth activation labels supervise training. Closed-loop runs do not receive them.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Dataset. Bench2Drive evaluates closed-loop driving on short interactive routes rather than on one long route whose score is dominated by a single infraction (Jia et al., 2024). We use its base split: 950 clips for training and 50 clips for open-loop validation, covering 44 scenario types, together with the 220 closed-loop test routes. **Metrics.** Closed-loop driving reports Driving Score (DS) and Success Rate (SR) (Jia et al., 2024). DS averages route completion after an infraction penalty. SR is the fraction of routes completed with no infraction. Open-loop planning is the average L2 displacement over two seconds at 2 Hz on the 50 validation clips. CondAD does not report efficiency or comfortness. The same 220 routes are also grouped into Merging, Overtaking, Emergency Brake, Give Way, and Traffic Sign. A route may belong to more than one group (Appendix A.1).

Implementation. The planner uses a ResNet-50 image backbone and an FPN neck. All six surround cameras are resized to 640×352. Detection, map, and planning queries follow the HiP-AD decoder; CondAD adds the yield query and the instruction query from Section 3. Identity verification is a separate stage: a frozen ResNet-18 classifier is trained on detection crops and then held fixed while the planner trains. Planner optimization uses AdamW with cosine annealing for four epochs. Training and closed-loop evaluation use four NVIDIA A100 GPUs with 40 GB of memory each. Crop size, learning rates, and the per-GPU batch are in Appendix A.2. Open-loop displacement is measured on the 50-clip validation split. Each comparison row in Table 1 except Ours is the published Bench2Drive result from the paper cited for that method, not a rerun in our codebase. When a later paper reprints an earlier baseline with a different score, we keep the earlier source.

4.2 OPEN-LOOP AND CLOSED-LOOP PLANNING

Protocol. Table 1 is the Bench2Drive comparison: open-loop displacement on the 50 validation clips, and closed-loop driving score and success rate. DriveTransformer-L has the smallest L2. CondAD's open-loop L2 is 0.71, its driving score is 85.48, and its success rate is 71.56. The rate 71.56 is 156/218 because this run finished 218 of the 220 routes.

Table 1: Open-loop and closed-loop planning on Bench2Drive. Avg. L2 is the two-second, 2 Hz displacement. Bold marks the best value among methods reported on the base split. A dash means that source does not report the entry. Ours success rate is 156/218 routes (71.56%), not a 220-route rate.

	Open-loop	Closed-loop	
Method	Avg. L2 ↓	DS ↑	SR ↑
UniAD-Base (Hu et al., 2023)	0.73	45.81	16.36
VAD (Jiang et al., 2023)	0.91	42.35	15.00
SparseDrive (Sun et al., 2025)	0.87	44.54	16.71
MomAD (Song et al., 2025)	0.82	47.91	18.11
DriveTransformer-L (Jia et al., 2025)	**0.62**	63.46	35.01
Hydra-NeXt (Li et al., 2025)	0.92	73.86	50.00
ORION (Fu et al., 2025)	0.68	77.74	54.62
CogDriver (Liu et al., 2026)	0.63	78.21	56.93
SimLingo (Renz et al., 2025)	–	85.07	67.27
HiP-AD (Tang et al., 2025)	0.69	86.77	69.09
SparseDriveV2 (Sun et al., 2026)	–	**89.15**	70.00
TFv6 (Nguyen et al., 2026)	–	91.6	79.5
Ours	0.71	85.48	**71.56**

SimLingo and TFv6 are trained on other sets, so a higher score there is not marked as the best base-split result.

4.3 MULTI-ABILITY EVALUATION

Five skills. Table 2 scores the same 220 routes by skill. Because the groups overlap, these rates are not a split of the driving score in Table 1.

Table 2: Multi-ability closed-loop scores on Bench2Drive (%). Bold marks the best reported value. A dash means that source does not report the entry.

Method	Merge	Over.	Brake	Give	Sign	Mean
UniAD-Base (Hu et al., 2023)	14.10	17.78	21.67	10.00	14.21	15.55
VAD (Jiang et al., 2023)	8.11	24.44	18.64	20.00	19.15	18.07
DriveTransformer-L (Jia et al., 2025)	17.57	35.00	48.36	40.00	52.10	38.60
Hydra-NeXt (Li et al., 2025)	40.00	64.44	61.67	50.00	50.00	53.22
ORION (Fu et al., 2025)	25.00	71.11	78.33	30.00	69.15	54.72
SimLingo (Renz et al., 2025)	–	–	–	–	–	–
HiP-AD (Tang et al., 2025)	50.00	**84.44**	83.33	40.00	72.11	65.98
SparseDriveV2 (Sun et al., 2026)	**66.25**	75.55	75.00	50.00	71.57	67.67
Ours	50.00	**84.44**	**86.67**	**90.00**	**73.68**	**76.96**

Comparison with HiP-AD. The HiP-AD row is the published result (Tang et al., 2025). Against that row, merging stays at 50.00 and overtaking at 84.44. Emergency brake moves from 83.33 to 86.67, traffic sign from 72.11 to 73.68, and Give Way from 40.00 to 90.00 (9/10). The mean moves from 65.98 to 76.96. The gain is concentrated in Give Way. That column pools `YieldToEmergencyVehicle` and `InvadingTurn`, so it is not the emergency-vehicle subset (Jia et al., 2024).

4.4 ARCHITECTURE ABLATION

Table 3: Ablation of the yield trigger, the navigation command and target, and target persistence. Yield SR is the success rate on Bench2Drive's 10 Give Way routes. Trigger acc. is the per-frame rate at which the trigger matches a rear emergency vehicle, measured on the 220 routes.

Variant	Trigger	Nav command	Nav target	Target persistence	Yield SR ↑	Trigger acc. ↑
HiP-AD (Tang et al., 2025)	–	–	–	–	40.00	–
+ Trigger only	✓	–	–	–	40.00	**87.22**
+ Instruction pair	✓	✓	✓	–	50.00	86.32
Full CondAD	✓	✓	✓	✓	**90.00**	87.12

Condition factors. HiP-AD has none of the four factors. Trigger only adds activation supervision and does not build a navigation condition. The instruction pair turns that activation into a navigation command and a spatial target, but does not keep the target as the ego vehicle moves. Full CondAD also retains the target until the maneuver ends. On Bench2Drive's 10 Give Way routes, HiP-AD and trigger only both succeed on 40%, the instruction pair on 50%, and Full CondAD on 90% (9/10), compared with HiP-AD at 40.00%. Ground-truth activation labels y_t^* supervise training. Closed-loop runs do not receive them. Trigger accuracy is a separate per-frame score on all 220 routes: a frame is correct when the predicted category matches the rear emergency vehicle. HiP-AD has no trigger. The three triggered rows are close, at 87.22%, 86.32%, and 87.12%. A small difference is consistent with the yield maneuver changing the rear-camera viewpoint, not with a different recognizer. Figure 3 is that rear view on routes 3364 and 3373: the ambulance is still behind the ego vehicle when the yield starts.



Figure 3: **Rear-camera views on two closed-loop routes.** Routes 3364 and 3373 are shown for HiP-AD and for Ours at Trigger, Offset, In lane, and Passed. While the ambulance is behind the ego vehicle, the frame is the rear camera, and a red box labels that vehicle. A front-only view would miss an emergency vehicle approaching from behind. After it has passed, the Ours frames look forward at the back of the ambulance. Other vehicles carry blue boxes.

5 Conclusion

We presented CondAD, an interaction-conditioned end-to-end driving framework for reactive maneuvers whose navigation intent changes while the nominal route remains unchanged. Rather than relying on trajectory imitation alone to implicitly associate scene interactions with expert behavior, CondAD grounds the relevant actor–route interaction and converts it into an explicit command–target navigation condition. A maneuver state maintains this condition across planning steps while the continuous planner remains responsible for path and speed generation. This design follows the separation introduced in our method between interaction grounding, navigation specification, and trajectory execution.

On Bench2Drive, CondAD improves Give Way success from 40.00% to 90.00%. Emergency-vehicle yielding on the YieldToEmergencyVehicle subset rises from 20.00% to 80.00%. The progressive ablation further shows that trigger information alone does not improve Give Way performance in our setting, while introducing the command–target condition and subsequently maintaining the maneuver across planning steps yields successive improvements. These results support explicit, persistent navigation conditioning as an effective interface for interaction-driven closed-loop behavior, without replacing continuous trajectory learning.

AI Use Statement

AI assistance supported manuscript language refinement. The authors retain responsibility for the scientific content and final manuscript.

References

Zhili Chen, Maosheng Ye, Shuangjie Xu, Tongyi Cao, and Qifeng Chen. PPAD: Iterative interactions of prediction and planning for end-to-end autonomous driving. In Aleš Leonardis, Elisa

Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 239–256, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-72761-0.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. TransFuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12878–12895, 2023. doi: 10.1109/TPAMI.2022.3200245. URL https://ieeexplore.ieee.org/abstract/document/9863660.

Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Dingkang Liang, Chong Zhang, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. ORION: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24823–24834, October 2025.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17853–17862, June 2023.

Yi Huang, Zhan Qu, Lihui Jiang, Bingbing Liu, and Hongbo Zhang. Prioritizing perception-guided self-supervision: A new paradigm for causal modeling in end-to-end autonomous driving. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 49956–49982. Curran Associates, Inc., 2025. doi: 10.52202/085713-1666. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/476c9ac41c967332dd28fb844e166b34-Paper-Conference.pdf.

Zhiyu Huang, Haochen Liu, and Chen Lv. GameFormer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3903–3913, October 2023.

Xiaosong Jia, Yulu Gao, Li Chen, Junchi Yan, Patrick Langechuan Liu, and Hongyang Li. DriveAdapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7953–7963, October 2023a.

Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21983–21994, 2023b. URL https://openaccess.thecvf.com/content/CVPR2023/html/Jia_Think_Twice_Before_Driving_Towards_Scalable_Decoders_for_End-to-End_Autonomous_CVPR_2023_paper.html.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2Drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 819–844. Curran Associates, Inc., 2024. doi: 10.52202/079017-0025. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/017761f94a1cd66d01c041aff85492c4-Paper-Datasets_and_Benchmarks_Track.pdf.

Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. DriveTransformer: Unified transformer for scalable end-to-end autonomous driving. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 67227–67243, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/a7afc9957f1190223763b6ea93218f98-Paper-Conference.pdf.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. VAD: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8340–8350, October 2023.

Peidong Li and Dixiao Cui. Navigation-guided sparse scene representation for end-to-end autonomous driving. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 29118–29134, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/47cf6de08432fb54ffd6b44fc1260413-Paper-Conference.pdf.

Zhenxin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Zuxuan Wu, and Jose M. Alvarez. Hydra-NeXt: Robust closed-loop driving with open-loop training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27305–27314, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Li_Hydra-NeXt_Robust_Closed-Loop_Driving_with_Open-Loop_Training_ICCV_2025_paper.html.

Pei Liu, Qingtian Ning, Xinyan Lu, Haipeng Liu, Weiliang Ma, Dangen She, Xianpeng Lang, and Jun Ma. CogDriver: Integrating cognitive inertia for temporally coherent planning in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18150–18160, June 2026.

Long Nguyen, Micha Fauth, Bernhard Jaeger, Daniel Dauner, Maximilian Igl, Andreas Geiger, and Kashyap Chitta. LEAD: Minimizing learner-expert asymmetry in end-to-end driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 39775–39785, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Nguyen_LEAD_Minimizing_Learner-Expert_Asymmetry_in_End-to-End_Driving_CVPR_2026_paper.html.

Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. SimLingo: Vision-only closed-loop autonomous driving with language-action alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11993–12003, 2025. doi: 10.1109/CVPR52734.2025.01120. URL https://ieeexplore.ieee.org/abstract/document/11094118.

Ziying Song, Caiyan Jia, Lin Liu, Hongyu Pan, Yongchang Zhang, Junming Wang, Xingyu Zhang, Shaoqing Xu, Lei Yang, and Yadan Luo. Don't shake the wheel: Momentum-aware planning in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22432–22441, June 2025.

Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. SparseDrive: End-to-end autonomous driving via sparse scene representation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8795–8801, 2025. doi: 10.1109/ICRA55743.2025.11128800.

Wenchao Sun, Xuewu Lin, Keyu Chen, Zixiang Pei, Xiang Li, Yining Shi, and Sifa Zheng. SparseDriveV2: Scoring is all you need for end-to-end autonomous driving. In Paolo Favaro, Zuzana Kukelova, Atsuto Maki, Anna Rohrbach, Konrad Schindler, and Federico Tombari (eds.), *Computer Vision – ECCV 2026*, pp. 446–463, Cham, 2026. Springer Nature Switzerland. ISBN 978-3-032-37362-5.

Yingqi Tang, Zhuoran Xu, Zhaotie Meng, and Erkang Cheng. HiP-AD: Hierarchical and multi-granularity planning with deformable attention for autonomous driving in a single decoder. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25605–25615, October 2025.

Zebin Xing, Xingyu Zhang, Yang Hu, Bo Jiang, Tong He, Qian Zhang, Xiaoxiao Long, and Wei Yin. GoalFlow: Goal-driven flow matching for multimodal trajectories generation in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1602–1611, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Xing_GoalFlow_Goal-Driven_Flow_Matching_for_Multimodal_Trajectories_Generation_in_End-to-End_CVPR_2025_paper.html.

Yibo Yuan, Jianwu Fang, Yang Zhou, Zhao Yang, Chen Lv, and Jianru Xue. Causal-Planner: Causal interaction disentangling with episodic memory gating for autonomous planning. In

2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 335–341, 2025. doi: 10.1109/IROS60139.2025.11245887. URL https://ieeexplore.ieee.org/document/11245887.

Bozhou Zhang, Nan Song, Xin Jin, and Li Zhang. Bridging past and future: End-to-end autonomous driving with historical prediction and planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6854–6863, June 2025.

Julian Zimmerlin, Jens Beißwenger, Bernhard Jaeger, Andreas Geiger, and Kashyap Chitta. Hidden biases of end-to-end driving datasets, 2024. URL http://arxiv.org/abs/2412.09602.

A Supplementary Notes

A.1 Route groups

Bench2Drive scores the 220 closed-loop routes as a whole and again by skill. The skill groups are Merging, Overtaking, Emergency Brake, Give Way, and Traffic Sign. A route may belong to more than one group, so the five groups are not a partition of the 220. Give Way is ten routes, five `YieldToEmergencyVehicle` and five `InvadingTurn` (Jia et al., 2024). Yield SR in Table 3 is the success rate on that ten-route set.

A.2 Crops, optimization, and the reported trigger

Identity verification uses a frozen ResNet-18 on 128×128 crops. Each crop is cut from a detected box expanded by 15%. Only the linear classifier is trained, with AdamW at learning rate 10^{-3} and weight decay 10^{-2}, and is then held fixed. The planner uses AdamW at learning rate 2×10^{-4}, weight decay 10^{-3}, and a backbone learning-rate multiplier of 0.5. Its schedule is cosine annealing after a linear warmup. Training uses 12 samples on each GPU, in mixed precision, for four epochs.

Ground-truth activation labels y_t^* supervise training. Closed-loop runs do not receive them. The crop verifier in Section 3.2 is the identity check on the deployment path in Section 3.4.

A.3 Six-camera sequences

Figures 4–6 are the full storyboard for four closed-loop `YieldToEmergencyVehicle` routes: 3364, 3373, 3378, and 3380. Each route has seven time steps. A row is one time step, with the six surround cameras and a bird's-eye view. The rows run from the ambulance appearing behind the ego vehicle, through the offset and the side pass, to the ambulance having gone ahead. `CAM_BACK` is the view that contains the ambulance while it is still behind. The front cameras do not.

In the bird's-eye panel the black rectangle is the ego vehicle and the pink rectangle is the ambulance. The dark trace is the CondAD path up to the ego vehicle. The light-gray trace is the HiP-AD path on the same route. Routes 3364 and 3373 are the two routes in Figure 3.

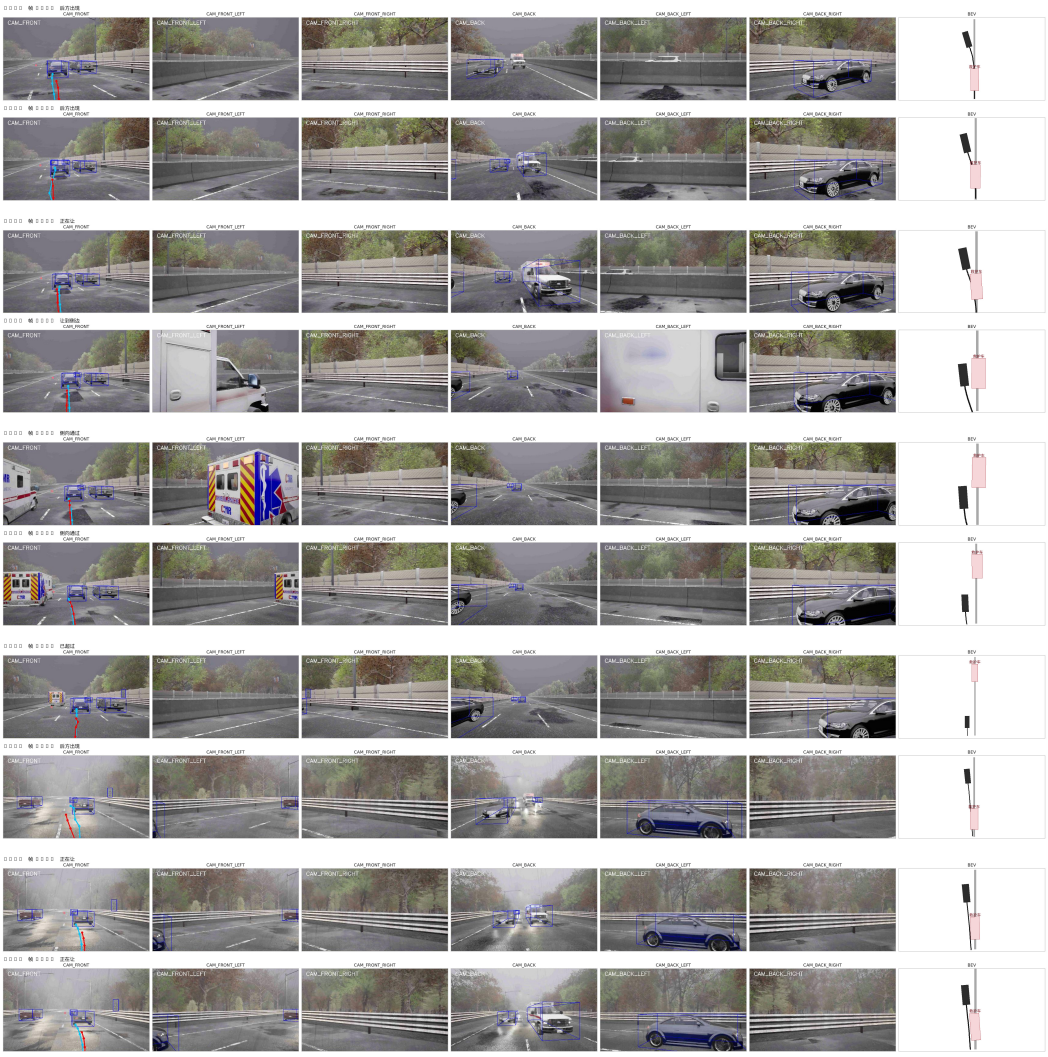

Figure 4: **Six cameras and a bird's-eye view, routes 3364 and 3373.** Each row is one time step. Camera order is `CAM_FRONT`, `CAM_FRONT_LEFT`, `CAM_FRONT_RIGHT`, `CAM_BACK`, `CAM_BACK_LEFT`, `CAM_BACK_RIGHT`, then BEV.

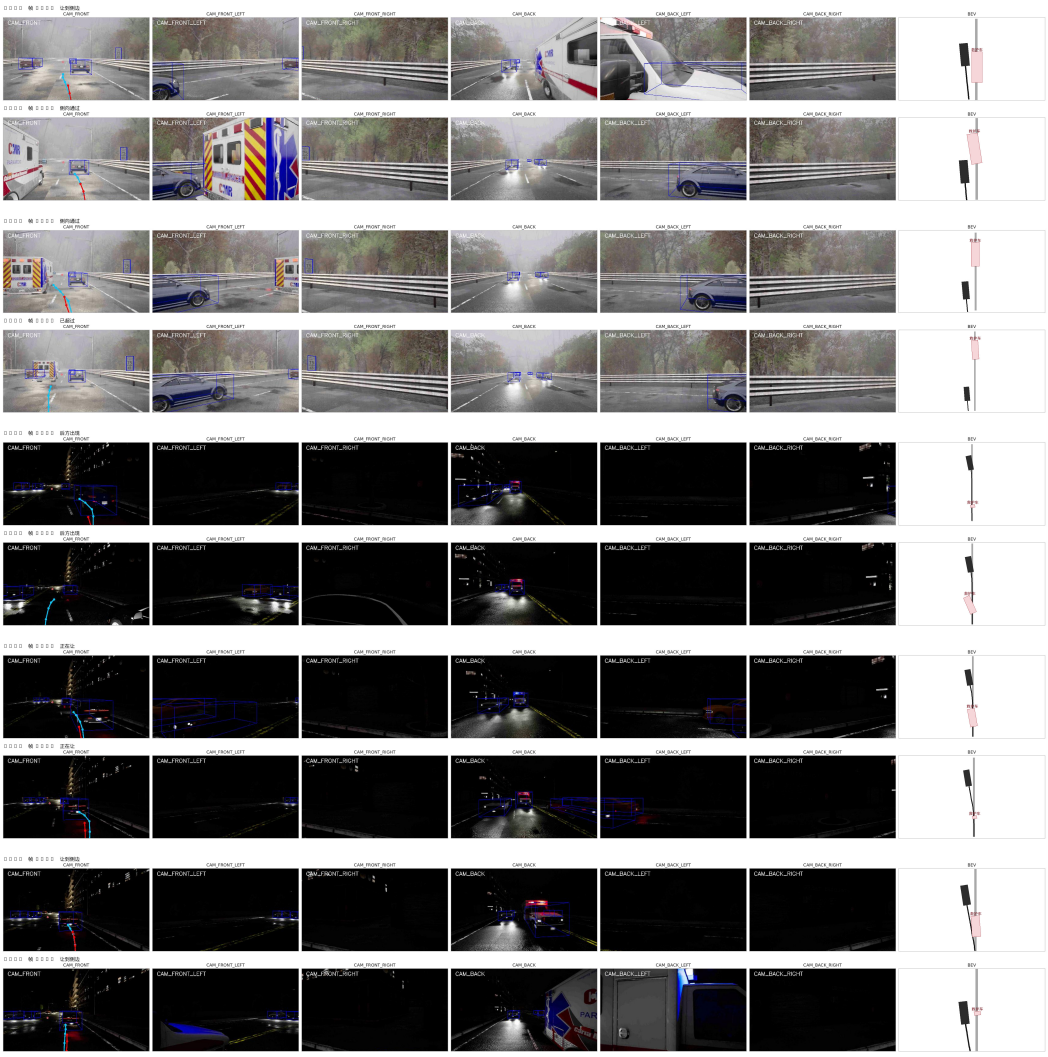

Figure 5: **Continued: routes 3373 and 3378.**

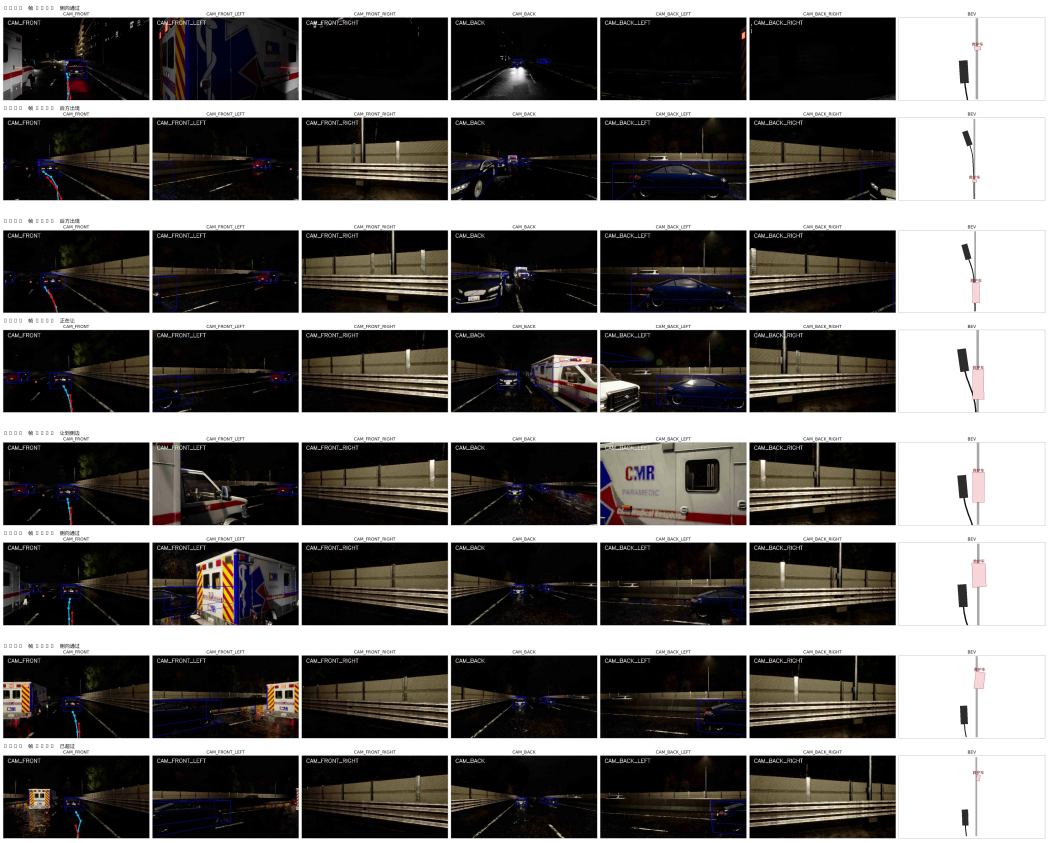

Figure 6: **Continued: routes 3378 and 3380.**