

Forgetting to Care, Forgetting to See: Interpreting ST-P3 Planning via LLM Explanations and Collision Visualizations

Antonio Sirica

s326811, Politecnico di Torino
Turin, Italy

ABSTRACT

We propose two novel concept-level intervention strategies to enhance interpretability and diagnostic capabilities in ST-P3, a model for autonomous driving. Leveraging a Concept Bottleneck Model (CBM) framework, we introduce interventions at two levels of the planning pipeline: (1) cost-level intervention, where we mask specific concept contributions within the cost map to simulate the model "forgetting" the cost of certain concepts (e.g., pedestrians or vehicles), while choosing the trajectory, and generate natural language explanations via a Large Language Model (LLM) that compares the current trajectory and the future chosen one costs to get explainable insights; and (2) perception-level intervention, where we mask out concepts directly in the segmentation map, making the planner unaware of their presence, and then produce visualizations that highlight resulting planning failures such as collisions. These complementary methods allow for both semantic and perceptual analyses of the planner's behavior under concept perturbations. Together, they demonstrate the value of CBMs not only for controllability but also for uncovering failure modes and guiding safer, more interpretable decision-making in autonomous systems.

KEYWORDS

XAI, CBM, AV, Visual Explanations, LLM-Based-Explanations, Olama

1 INTRODUCTION

Autonomous driving is one of the most complex applications of AI, requiring real-time perception, prediction, and control with high precision. While deep learning has driven major progress—especially through end-to-end learning—its black-box nature raises serious concerns. The lack of transparency in these systems hinders trust, complicates debugging, and poses challenges for legal accountability and regulatory approval[1].

Explainable AI (XAI) is critical for autonomous driving. This comprehensive review focuses on explainable-by-design (XbD) methods, which embed interpretability directly into AI architectures. Unlike post-hoc explanations, XbD aims for inherent transparency using semantic structures, attention mechanisms, auxiliary outputs, and interpretable representations.

We examine the architectural models of AVs—modular and end-to-end—and their interpretability challenges. Key XbD techniques include attention visualizations, semantic inputs, auxiliary tasks, natural language explanations, and causal analysis.

Finally, we highlight the ST-P3 framework, an interpretable, end-to-end architecture designed for robust and explainable driving. Building on ST-P3, our work introduces improvements to enhance explainability and real-world applicability. This review situates our contribution within the broader effort to develop trustworthy, transparent, and deployable autonomous driving systems.

Luca Pellicciotti

s331419, Politecnico di Torino
Turin, Italy

2 RELATED WORK

2.1 The Imperative for Explainable AI in Autonomous Driving

Advances in AI and deep learning have driven rapid progress in autonomous driving, promising safer and more efficient transportation. However, the opaque, black-box nature of many AI models remains a major barrier to trust, regulation, and accountability. Without clear insight into how AVs make decisions—especially in edge cases—public skepticism grows and legal compliance becomes difficult.

Explainable AI (XAI) is essential to address these concerns. Users need to understand AV behavior to build trust, particularly in unexpected or risky situations. From a sociotechnical perspective, aligning AV actions with human expectations requires interpretable systems [2].

Effective explanations must be forward-looking, clear to non-experts, and faithful to the system's true logic. They should help users and developers understand and anticipate decisions, not just interpret them after the fact.

Explainability is thus a core design challenge in autonomous driving, requiring interdisciplinary solutions.

2.2 Architectural Paradigms in Autonomous Driving: Modular vs. End-to-End

Autonomous driving systems are typically built using either modular pipelines or end-to-end learning. Modular designs split the driving task into components like perception, planning, and control. This structure supports clearer debugging and interpretability, as each module can be individually analyzed. However, it can suffer from inefficiencies, interface rigidity, and cascading errors.

End-to-end systems, by contrast, use deep learning to map raw sensor data directly to driving actions. They offer flexibility and potentially better performance, but their internal workings are often opaque, making it difficult to trace or explain decisions—a major hurdle for trust and accountability.

This trade-off between performance and transparency highlights the importance of explainable-by-design (XbD) methods. In end-to-end systems especially, embedding semantic structure, auxiliary outputs, and interpretable reasoning directly into the architecture is crucial. The following sections explore how these strategies aim to combine the power of deep learning with the need for explainability in real-world autonomous driving.

2.3 STP-3

ST-P3 is a vision-based end-to-end framework for autonomous driving that integrates perception, prediction, and planning within a unified architecture. Unlike traditional pipelines relying on HD maps, ST-P3 directly processes multi-view camera inputs to generate spatial-temporal features. The perception module employs an egocentric aligned accumulation technique, which aligns and

aggregates past and present features in 3D space, preserving geometric consistency before transforming them into the bird’s-eye view (BEV) representation. For prediction, a novel dual pathway design models motion variations in the past to enhance future semantic predictions. Finally, the planning module incorporates prior knowledge early in the network and refines the planned trajectory using a cost volume and command-conditioned sampler. ST-P3 demonstrates state-of-the-art performance on benchmarks such as nuScenes and CARLA, highlighting its effectiveness as a fully vision-based alternative to LiDAR-centric approaches.

2.4 Explainable-by-Design Methodologies

As post-hoc methods often fail to deliver reliable, actionable insight, explainable-by-design (XbD) approaches have emerged as a powerful alternative. These methods incorporate interpretability directly into the system architecture, offering explanations that are faithful by construction. We organize this discussion by methodological categories, detailing key models and how they compare to **ST-P3** (Hu et al., 2022), our reference vision-based framework.

2.4.1 Attention-Based Architectures. A large class of XbD approaches exploits attention mechanisms to reveal input regions influencing control outputs.

The **Causal Attention Model** by Kim and Canny (2017) was among the first to apply causal filtering to saliency maps, removing spurious attention blobs and emphasizing temporally consistent features (e.g., lane markings, vehicles).

Object-Centric Visual Attention, introduced by Wang et al. (2019), directed attention over object proposals generated by a pre-trained detector, focusing on semantically meaningful elements within the global scene context.

In **End-to-End Visual Attention**, Cultrera et al. (2020) proposed a conditional imitation learning model with a trainable attention mechanism that localizes image regions most responsible for a given driving decision.

More recently, **Driver Attention for Explainable Decision-Making** (Araluce et al., 2024) used driver gaze data in conjunction with ARA-GAN and MobileNetV2 to produce attention maps grounded in real human visual behavior.

While these models offer useful visual cues, attention weights often lack faithfulness and are sensitive to perturbations. In contrast, ST-P3 embeds interpretability through modular components—explicitly modeling perception, prediction, and planning—which reduces reliance on saliency maps and enhances causal traceability through spatial-temporal features.

2.4.2 Semantic Inputs and Auxiliary Objectives. Another family of models enhances interpretability by integrating semantic priors or auxiliary outputs. **LiDAR-based BEV Maps** by Caltagirone et al. (2017) used 3D point clouds projected into top-down occupancy grids, facilitating spatial understanding but lacking class-specific semantics.

The **Affordance and Action Primitive** model by Mehta et al. (2018) introduced auxiliary tasks such as predicting affordances and abstract driving commands to increase transparency and training stability.

ChauffeurNet (Bansal et al., 2019) made interpretability a core feature by stacking rich semantic maps (traffic lights, agents, lanes) as multi-channel inputs, enabling explicit interventions and counterfactual analysis.

Multi-modal Trajectory Prediction (MTP) by Cui et al. (2019) generated BEV semantic maps that encoded multiple future outcomes by class, improving spatial diversity and interpretability.

Neural Motion Planner (NMP) from Zeng et al. (2019) generated intermediate representations—such as object detections, predicted paths, and cost maps—interpretable by design.

ST-P3 distinguishes itself from these approaches by not requiring pre-labeled semantic maps or LiDAR. Its *Egocentric Aligned Accumulation* preserves 3D spatial consistency before BEV projection, and its *Dual Pathway* design explicitly models temporal uncertainty in agent trajectories.

2.4.3 Natural Language Generation. Some XbD efforts prioritize human-readable explanations via text. **Video-to-Text Explanation** (Kim et al., 2018) translated attention-weighted video frames into short textual justifications. **Explainable Driving with Language Grounding** (Hendricks et al., 2018) combined visual and textual representations to produce multi-modal explanations for driving actions.

Although these systems enhance human accessibility, they rely on auxiliary language modules and often fail to align tightly with model internals. ST-P3 avoids post-hoc textual generation by exposing interpretable intermediate outputs—like semantic maps and trajectory probabilities—directly tied to the decision-making process.

2.4.4 Spatiotemporal Causal Modeling. Temporal abstraction plays a key role in several recent XbD approaches. **STP (Spatio-Temporal Perception)** (2021) [3] demonstrated how temporal cues can reveal abstract concepts like driver distraction through causal video decoding.

ST-P3 [4] builds on this trend by aligning egocentric features across time, generating probabilistic forecasts for both ego and agent motion, and refining control through a cost-based, command-conditioned planner.

The **BridgeAD** model (2025)[5] introduced the idea of latent belief states—structured abstractions that evolve over time and mediate between perception and planning layers. These latent variables encode both observed and inferred factors, supporting temporally grounded explanations.

2.4.5 Explanation Robustness and Stability. A complementary goal to interpretability is explanation reliability under perturbation. **DRIVE** (Lai et al., 2024)[6] introduced a framework focused on this dimension. By defining four properties—*consistent interpretability*, *stable interpretability*, *consistent output*, and *stable output*—DRIVE addresses the instability observed in previous concept bottleneck models like DCG. It proposes methods to stabilize explanations and predictions under noisy or shifted inputs.

ST-P3 and DRIVE target distinct but complementary goals: while ST-P3 embeds structured, modular transparency, DRIVE ensures those explanations remain valid under real-world uncertainty. Future research could benefit from integrating both approaches to build AV systems that are both interpretable and dependable.

2.4.6 Concept Bottleneck Models. Concept Bottleneck Models (CBMs) [7] predict decisions in two stages: first identifying human-interpretable concepts, then using them to make final predictions. This approach improves transparency by exposing intermediate reasoning steps, making models easier to understand, audit, and debug. CBMs are especially valuable in high-stakes areas like autonomous driving,

where accountability matters. They also support human-in-the-loop corrections, enhancing trust and reliability in AI systems.

2.4.7 Causal Concept Effect (CaCE). The Causal Concept Effect (CaCE) [8] is a principled metric for quantifying the causal influence of semantically meaningful intermediate concepts—such as those predicted by Concept Bottleneck Models—on system outputs. Unlike correlation-based measures, CaCE uses interventionist causal techniques grounded in do-calculus to assess how deliberate changes to specific concepts (e.g., pedestrian presence, traffic light status) impact control actions or trajectories.

By isolating true causal effects, CaCE distinguishes relevant concepts from spurious correlations, supporting more robust and trustworthy explanations. It aids model auditing and promotes training toward causally grounded and transparent decision-making.

2.5 nuScenes

The nuScenes dataset [9] is a large-scale, multimodal benchmark for autonomous driving. Collected by Aptiv (now Motional) in Boston and Singapore, it features over 1,000 diverse 20-second driving scenes at 2 Hz, captured via six cameras, five radars, and a 32-beam LiDAR. The dataset includes high-quality 3D annotations, detailed maps, and supports spatiotemporal modeling through its sequential structure.

nuScenes also provides CAN bus data with time-synchronized vehicle state information (e.g., speed, acceleration, steering angle), enabling precise ego-motion modeling, validation of control outputs, and development of physics-informed planning algorithms.

3 RESEARCH GAPS

Despite growing interest in explainability for autonomous driving, most current methods rely on post-hoc analyses, offering limited insight into the causal impact of high-level concepts on driving decisions. Few approaches allow active, concept-level interventions within end-to-end planners to test how semantic information influences behavior.

While Concept Bottleneck Models (CBMs) show promise in classification, their integration into complex planners like ST-P3—especially for natural language and visual explanations—remains underexplored. Tools to simulate semantic failures (e.g., missing pedestrians) and quantify their effect on decisions are also lacking.

Moreover, existing frameworks rarely offer multi-level interpretability across both perception and cost computation stages. Most explainable-by-design models operate at the frame level, ignoring how decisions evolve over time. This limits understanding of temporal causality—crucial in real-world driving, where past context shapes current actions.

Altogether, there is a pressing need for intervention-based methods that support spatial and temporal concept-level reasoning, enabling more robust, transparent, and trustworthy autonomous systems.

4 METHODOLOGY

4.1 Intervention via Concept Bottleneck Framework

To address these gaps, we introduce two intervention mechanisms grounded in the Concept Bottleneck framework. These mechanisms are applied specifically within the planning module of the ST-P3 architecture. Crucially, our method operates entirely at inference time: it does not modify the perception or prediction components,

nor does it require retraining the model. Instead, we apply these interventions on a curated subset of the nuScenes dataset, enabling controlled experiments with minimal system disruption.

The following describes in details the two different approaches.

4.2 "Forgetting to Care"

4.2.1 Concept Masking. In our case, ST-P3 already predicts a range of interpretable semantic maps, such as pedestrians, drivable areas, and lane dividers. These concepts are used internally by the rule-based components of the cost function to evaluate the feasibility and safety of candidate trajectories by computing the specific cost referred to the concepts (there are also other costs that are not segmentation-dependant). By inserting a binary control mechanism, where each concept can be switched on or off, we reframe this cost function as a modular Concept Bottleneck. This allows us to isolate and evaluate the causal impact of individual concepts on the planner’s behavior.

This architecture uses a modular cost function to guide trajectory selection, with each component addressing a specific safety or compliance factor. In particular, by deselecting a concept, so a cost, we force the model to choose the best trajectory without caring of that specific cost. The safety cost penalizes predicted collisions, especially at high speeds, based on obstacle predictions from semantic maps. The headway cost maintains safe following distances by discouraging trajectories that come too close of detected vehicles.

The lane divider cost penalizes proximity to lane boundaries, discouraging unsafe or illegal lane changes using fine-grained segmentation of lane markings. The cost volume, unlike rule-based components, is learned from data and encodes preferences for comfort, efficiency, and safety based on scene context. The rule cost enforces legal constraints by penalizing paths that leave the drivable area.

In addition to the semantic concepts we analyze, there are also Comfort and Progress costs, which are not treated as deselectable components in our study. However, they are still considered during the trajectory selection process in planning. The Comfort cost penalizes trajectories that exhibit high lateral acceleration, jerk, or curvature, promoting smoother motion. The Progress cost, on the other hand, encourages trajectories that efficiently move the vehicle toward its destination, favoring forward advancement.

A Concept Bottleneck mechanism applies binary masks to each cost term, enabling analysis of planner behavior under simulated perception failures. By disabling specific costs—e.g., for pedestrians or lane dividers—we can test whether the planner still behaves safely and legally, revealing its dependency on individual semantic concepts costs.

4.2.2 Current Trajectory: Physics-Grounded Baseline. To provide a baseline for analysis, we compute a deterministic reference trajectory, termed the *current direction trajectory*. This is derived analytically from the vehicle’s current speed and steering angle—parameters retrieved from the nuScenes CAN bus data. When curvature is low, the vehicle is assumed to continue straight; otherwise, it follows a circular arc defined by its turning radius. The resulting trajectory represents the ego vehicle’s physical path under constant motion inputs and serves as a physics-grounded comparison for understanding how model-generated trajectories deviate.

4.2.3 Cost Normalization. To ensure comparability across different cost components and trajectory samples, we apply min-max normalization to the aggregated costs. For each cost term (e.g., safety, progress), the costs are first aggregated over the future time horizon

for all candidate trajectories. Then, for each batch, the minimum and maximum values across the full set of trajectories are computed. The costs of the selected best and worst trajectories are normalized as follows:

$$\text{Normalized Cost} = \frac{\text{Cost} - \min(\text{Cost})}{\max(\text{Cost}) - \min(\text{Cost}) + \epsilon}$$

where ϵ is a small constant added for numerical stability. This normalization scales each cost to the range $[0, 1]$, enabling fair comparison between different cost components and improving the interpretability of the results.

Also the current trajectory is normalized in function of all the trajectory costs.

4.2.4 Visual Comparative Evaluation and LLM-based Explanation.

After concept ablation, we generate visual explanations of the selected trajectory, highlighting potential collisions.

By comparing this reference trajectory to the planner's output under different concept ablations, natural language explanations are produced by an LLM—in our case, the lightweight Ollama-based model *Mistral*, which supports real-time generation of insights. Its proven real-time performance is demonstrated through partnerships with automotive leaders like Stellantis [10].

In doing so, our approach bridges the gap between interpretable AI frameworks and state-of-the-art end-to-end driving models, delivering actionable insights into the internal logic of trajectory planning systems.

4.3 "Forgetting to See"

In this section, we aim to evaluate the causal influence of specific semantic concepts—namely *pedestrians* and *vehicles*—on the model's trajectory selection, by "hiding" concepts at a different level of the planning. To this end, we adapt the Causal Concept Effect (CaCE) methodology to our driving setup. This approach allows us to test how sensitive the trajectory predictions are to the presence or absence of certain concepts in the input map.

4.3.1 Concept Representation and Interventions. The planning module receives as input a semantic map that is internally represented as a spatial occupancy matrix. Each cell in this matrix encodes the presence or absence of specific scene elements (e.g., vehicles, pedestrians, road boundaries) identified during the segmentation process. These elements are encoded as vectors at specific coordinates in the grid. Larger objects, such as trucks or buses, span multiple adjacent coordinates and thus appear as clusters of occupied cells.

To simulate counterfactual scenarios where a concept is removed, we perform targeted interventions by zeroing out all vectors corresponding to that concept across the map. For example, to assess the effect of pedestrians, all occupancy vectors associated with pedestrian coordinates are set to zero, effectively removing them from the model's perception.

4.3.2 Trajectory Comparison and Collision Analysis. We then feed these altered maps to the model and compare the resulting trajectories with the baseline predictions produced from the unmodified input. For each case, we overlay the original and intervened trajectories on the same semantic map to enable visual inspection and register the possible collision.

4.3.3 Explainability via Visual Analysis. Rather than quantifying the effect of concept removal through numerical metrics, we opted for a graphical representation of the results. This choice was driven

by the goal of enhancing explainability: visualizing both the original and modified trajectories on the same map allows for intuitive interpretation of the consequences of each intervention. For instance, it becomes immediately clear when the absence of a concept leads to riskier paths or potential collisions, which may not be as evident from abstract numerical summaries.

5 EXPERIMENTS AND ANALYSIS

5.0.1 Protocols. For the experiments, we performed inference using the ST-P3 model, with the pretrained weights given by the repository github.com/OpenDriveLab/ST-P3. The model was used with its default configuration on the nuScenes mini dataset. The model processed inputs from six surround-view cameras, resized to 224×480 resolution, using 3 past frames to predict 4 future steps. Bird's-eye view (BEV) features were generated over a 100×100 m grid with 0.5 m resolution, without relying on HD maps. The EfficientNet-b4 encoder and a temporal block with ego-motion were used to extract spatiotemporal features. Planning was enabled with 600 sampled trajectories and high-level command inputs. Evaluation focused on semantic segmentation, future prediction, and planned trajectories using default BEV-based visualizations.

5.0.2 Collisions Representation. To detect collisions resulting from interventions, we compare the predicted trajectory coordinates with the occupancy matrix. If any point along the trajectory intersects a previously occupied cell (e.g., a vehicle or pedestrian location), we consider this a collision. These collisions are marked with a black "X" at the location of impact.

5.0.3 Hardware. For our analysis, we utilized the NVIDIA T4 GPU available in the free tier of Google Colab.

5.0.4 Materials. All scripts necessary to reproduce the experiments are available at: <https://github.com/Asir29/XAI-for-ST-P3>. Specifically, the scripts located in the "XAI-Scripts" directory are optimized for Google Colab and can be executed directly without modifications; the README file explains how to play with the experiments. Given the lack of space, more experiments can be found in the Appendix section.

5.0.5 "Forgetting to Care", Results.

Collision Visualizations. In the experiments, we show the surround view of the current frame alongside the bird's-eye view (BEV), which includes segmented instances as described in the legend. These visualizations allow us to analyze how the deactivation of specific semantic concepts influences the selected trajectory. Both the ground truth and the planned trajectory are shown. We also highlight potential collisions with a black cross when the trajectory intersects any segmented instance.

In Figure 1, we observe that deactivating the drivable area cost results in a trajectory that covers more space and exhibits higher speed. The no rule configuration allows the trajectory to slightly drift outside the drivable area. When the cost volume is removed, the resulting trajectory is more similar to the ground truth—possibly because it avoids overfitting to specific learned patterns. Finally, the configuration with all costs deactivated produces a trajectory that spans a significantly larger area and leads to a likely collision with a lane divider.

Figure 2 presents a representative example where the model must plan a fast and reliable trajectory to avoid a collision with a vehicle ahead. As expected, when the obstacle cost is deactivated, the

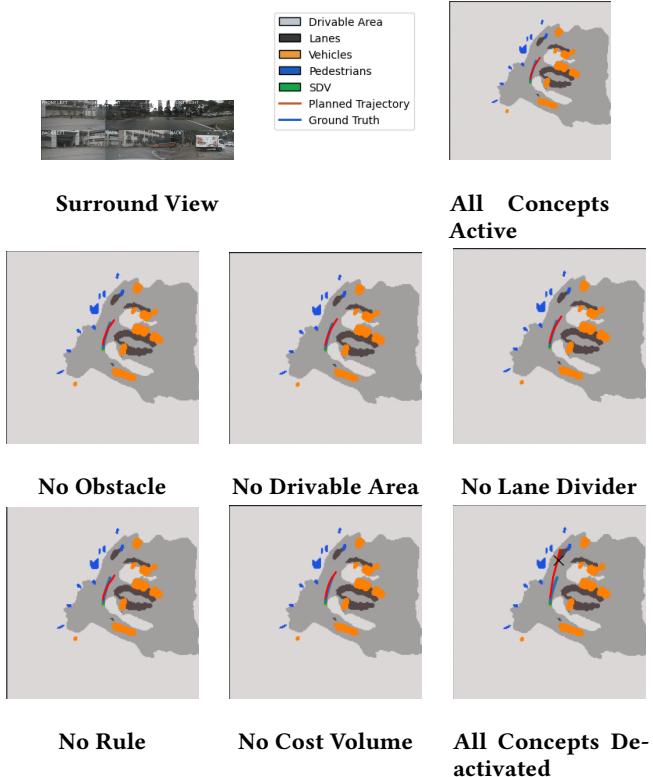


Figure 1: Comparison of planning results with individual concept deactivations (Scenario 2), using the same base trajectory. When a concept is deactivated, its associated cost is ignored during trajectory selection.

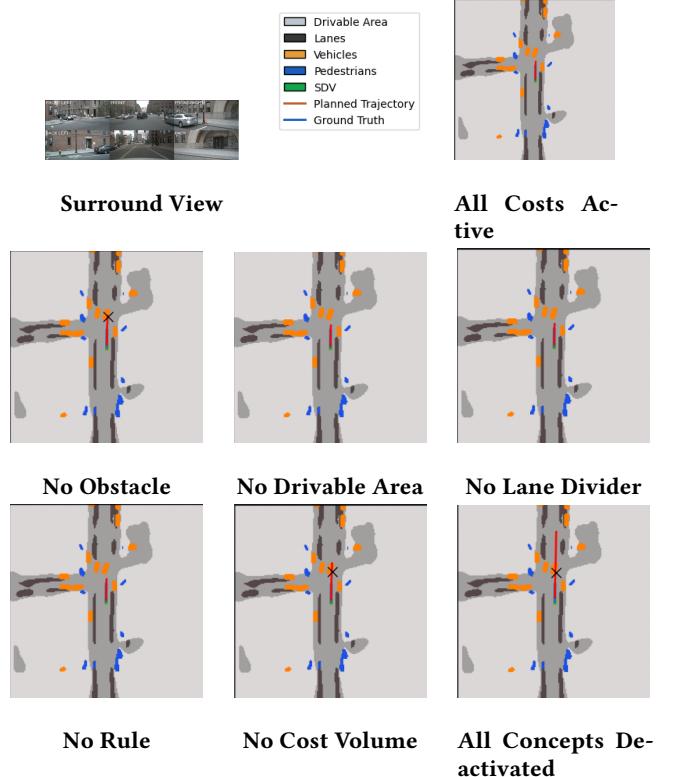


Figure 2: Comparison of planning results with individual concept deactivations (Scenario 50), using the same base trajectory. When a concept is deactivated, its associated cost is ignored during trajectory selection.

model selects a trajectory that covers excessive space, significantly increasing the risk of collision.

In this scenario, the Drivable Area and Lane Divider concepts have limited influence on the planning outcome.

Interestingly, the Cost Volume—which captures implicit knowledge learned from training data—also plays a crucial role. When removed, the trajectory becomes less safe and less aligned with expected driving behavior. Finally, when all costs are deactivated, the model produces an erratic, seemingly random trajectory, underscoring the importance of the combined cost structure for coherent and safe planning.

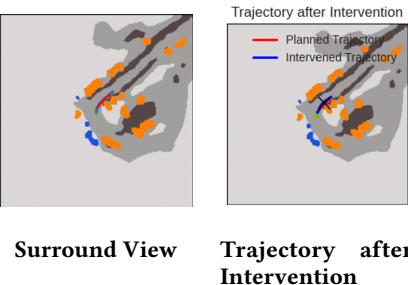


Figure 3: Trajectory comparison after concept removal (Image 14). The original trajectory avoids a parked vehicle, while the modified one collides with it due to the absence of vehicle information in the semantic map.

LLM Explanations. The LLM-based insights focus on analyzing the current vehicle direction in relation to the selected future trajectory, aiming to identify which semantic concepts influenced the model's decision-making during planning.

Below is an example of an LLM-generated explanation (for the case where all costs are active) for the Figure 1:

"1. The planned trajectory was chosen over the current one primarily due to its higher Progress cost (0.6532 vs 0.9901), indicating a

better outcome in terms of efficiency and progress on the intended route.

2. In making this decision, the model heavily weighted the Progress cost, as it seeks to optimize for efficient progress along the desired path.
3. The safety, headway, lane divider, rule, cost volume, comfort costs were given less importance in this instance since they all had lower values in the planned trajectory compared to the current one, indicating that these factors are less critical at this moment.
4. In this case, the progress of the vehicle was prioritized over other factors."

In this scenario, the LLM insight identifies Progress as the dominant factor influencing the decision. It recognizes that the planning module assigned significant weight to efficient forward movement, deeming it more important than other cost dimensions such as safety or comfort at that specific moment.

In the following the LLM insights for the Figure 2, in this case **de-selecting the Headway concept**:

- "1. The planned trajectory was chosen over the current one primarily due to the higher Progress cost (0.7835 vs 0.9164), indicating it is likely to reach the destination sooner or more efficiently.
2. In this case, the model appears to have prioritized the progress of reaching the destination over safety considerations, as the Safety cost for the planned trajectory was significantly higher (0.0615 vs 0.0064).
3. It is important to note that the model's decision is based on a balance between various costs, including Progress, Safety, and others, with the exact weightings learned from training data rather than fixed rules. In this specific instance, Progress was given more importance than Safety.
4. The Cost Volume, Comfort, Headway, Lane divider, and Rule costs had minimal impact on the decision as they were either identical or negligible in both trajectories (0.0000 for all but Progress, Safety, and their corresponding costs)."

Interestingly, this case highlights a potential failure mode: in the absence of the Headway cost, the model may overemphasize Progress, even when it leads to unsafe behavior. This suggests that headway awareness is not only essential for safety but also acts as a counterbalance to overly aggressive planning, emphasizing the need for proper concept weighting in critical scenarios.

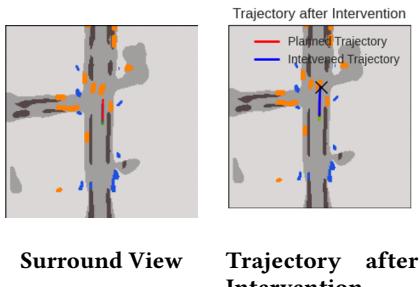


Figure 4: Trajectory comparison after concept removal (Image 50). The original planner slows down to yield to a turning vehicle ahead, whereas the modified version proceeds straight and causes a frontal collision.

5.0.6 "Forgetting to See", Results.

Collision Visualization. Removing the *pedestrian* and *vehicle* concepts from the semantic occupancy map revealed how sensitive the planner is to the presence of these elements. In the majority of the test cases, a change in the planned trajectory was observed after the intervention, but the difference was often minor or negligible. This was primarily due to two factors: either the planner already generated a failing trajectory in the original setting (e.g., resulting in a collision), or there were no vehicles or pedestrians between the ego vehicle and the target point to influence the path.

The most significant changes occurred in two key examples—Figures 3 and 4—where the intervention caused a trajectory that led to a collision. In Figure 3, the original trajectory correctly avoided a parked car, while the modified one collided with it due to the absence of the vehicle in the map. In Figure 4, the planner initially slowed down to account for a turning car ahead, but after the removal of vehicles and pedestrians, it no longer perceived the obstacle and continued forward, causing a frontal collision.

These results indicate that concept removal tends to have a stronger impact in scenes with a high density of semantic elements like cars and pedestrians. In such cases, their absence deprives the planner of critical context, leading to riskier or unsafe behaviors. The findings suggest that while the planner is often robust to visual perturbations, its reliance on key concepts becomes evident in more complex scenarios.

Although LLM-based explanation could in principle be applied to this intervention as well—for instance, to verbalize the causal role of removed concepts—it was not included here in order to keep the two explainability methods (causal interventions and language-based reasoning) conceptually distinct and separately interpretable.

6 CONCLUSIONS

The experiments show that concept-level interventions—at both cost and perception stages—help uncover how the ST-P3 model makes decisions and where it can fail. By masking specific concepts on the cost map and segmentation map, we generated natural language explanations and visualized critical planning failures like collisions. These methods using Concept Bottleneck Models make the model's reasoning transparent, reveal its weak points, and support safer, more interpretable autonomous driving. Overall, our results confirm that such explainable-by-design strategies are effective for building trustworthy and reliable AI systems in self-driving vehicles.

REFERENCES

- [1] Éloi Zablocki, H. Ben-Younes, P. Pérez, and M. Cord, “Explainability of deep vision-based autonomous driving systems: Review and challenges,” 2022. [Online]. Available: <https://arxiv.org/abs/2101.05307>
- [2] S. Atakishiyev, M. Salameh, H. Yao, and R. Goebel, “Explainable artificial intelligence for autonomous driving: A comprehensive overview and field guide for future research directions,” 2024. [Online]. Available: <https://arxiv.org/abs/2112.11561>
- [3] Q. Chang, W. Dai, Z. Shuai, L. Yu, and Y. Yue, “Spatial-temporal perception with causal inference for naturalistic driving action recognition,” 2025. [Online]. Available: <https://arxiv.org/abs/2503.04078>
- [4] S. Hu, L. Chen, P. Wu, H. Li, J. Yan, and D. Tao, “St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning,” 2022. [Online]. Available: <https://arxiv.org/abs/2207.07601>
- [5] B. Zhang, N. Song, X. Jin, and L. Zhang, “Bridging past and future: End-to-end autonomous driving with historical prediction and planning,” in *CVPR*, 2025.
- [6] S. Lai, T. Xue, H. Xiao, L. Hu, J. Wu, N. Feng, R. Guan, H. Liao, Z. Li, and Y. Yue, “Drive: Dependable robust interpretable visionary ensemble framework in autonomous driving,” 2024. [Online]. Available: <https://arxiv.org/abs/2409.10330>

- [7] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang, "Concept bottleneck models," 2020. [Online]. Available: <https://arxiv.org/abs/2007.04612>
- [8] Y. Goyal, A. Feder, U. Shalit, and B. Kim, "Explaining classifiers with causal concept effect (cace)," 2020. [Online]. Available: <https://arxiv.org/abs/1907.07165>
- [9] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Lioung, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," 2020. [Online]. Available: <https://arxiv.org/abs/1903.11027>
- [10] R. Online. (2025, Feb.) Gruppo stellantis: i nuovi progetti con mistral ai. Accessed: 2025-07-07. [Online]. Available: https://www.quattroruote.it/news/industria-finanza/2025/02/07/gruppo-stellantis-partnership_e_nuovi_progetti_con_mistral_ai_.html

APPENDIX: EXTENDED EXPERIMENTAL RESULTS

In this appendix, we present additional experiments and results that complement the main findings discussed in the Experiments and Analysis section. These include significant visual explanations and LLM-generated insights that could not be included in the main text due to space constraints.

.1 Extended "Forgetting to Care" Experiments and Analysis

This section provides a detailed analysis of planning results when individual semantic concepts are deactivated. These concepts include obstacle detection, drivable area recognition, lane dividers, traffic rules, and cost volume.

When all semantic concepts are active, the selected trajectory adheres to expected safety and driving norms. However, as individual costs are removed from the planning process, deviations in the trajectory reveal the influence and relative importance of each concept. Notably, the removal of the cost volume leads to a trajectory that results in boundary collisions, emphasizing the role of this learned prior in guiding safe behavior.

Collision Visualization. In Figure 5, we observe that when all semantic concepts (obstacle, drivable area, lane divider, traffic rules) are active, the trajectory shows only a minor deviation, indicating these constraints are not strongly conflicting in this scenario.

However, when the cost volume is removed—along with all other concepts—the resulting trajectory collides with the lane boundaries. Since the cost volume is derived from training data, this outcome highlights its essential role in encoding prior knowledge and steering the model toward safe and rule-compliant planning.

In Figure 6, we observe a particularly interesting scenario where the autonomous vehicle must navigate through a narrow street. Even with all semantic concepts active, the model plans a trajectory that appears risky—bringing the vehicle uncomfortably close to potential lane or vehicle collisions. This highlights the inherent complexity of the scene, where strict adherence to all constraints still results in a high-stakes maneuver.

What's especially noteworthy is that deactivating most individual concepts (e.g., obstacle, lane divider, drivable area) does not significantly change the outcome—the model continues to select similarly aggressive paths, still risking boundary violations. This consistency suggests that no single concept dominates the planning process here; rather, their combined influence reinforces a trajectory that the model deems viable under pressure.

However, when the cost volume is removed—a component representing what the model has implicitly learned from training data—the planner takes a markedly different approach. The new trajectory becomes more conservative, with slower movement and

reduced spatial coverage. This behavior indicates a cautious fallback strategy, where in the absence of learned priors, the model prioritizes safety over progress.

In Figure 7, it is notable that when all cost components are active, the model selects a trajectory that is significantly slower than the ground truth. This conservative behavior suggests an overemphasis on safety-related costs. Among the individual cost terms, the cost volume appears to be the most critical for ensuring safety: when it is removed, the model opts for a faster trajectory relative to the ground truth, which brings it dangerously close to the vehicle on the left-forward side, increasing the risk of collision.

This highlights a key trade-off in trajectory planning between safety and efficiency, and suggests that the cost volume plays a pivotal role in encoding spatial awareness of nearby dynamic agents. Its absence not only reduces caution but also undermines the model's ability to reason about occlusions and potential conflicts in complex traffic scenarios.

LLM Explanation. Regarding Figure 5, the LLM provides the following explanation when the lane divider concept is deactivated:

"1. The primary reason for choosing the planned trajectory over the current one is the significantly lower 'progress' cost (0.7498 vs 0.9728). This indicates that the planned path reaches the goal more efficiently.

2. The planned trajectory has a higher 'lrddivider' cost (0.3619), meaning it is closer to lane boundaries. This suggests a trade-off: the model prioritized faster arrival at the cost of some lane compliance.

3. The 'comfort' cost is zero for the planned trajectory and non-zero (0.1697) for the current one, indicating a preference for smoother, more comfortable driving.

4. Costs such as 'safety', 'headway', 'rule', and 'costvolume' were negligible or equal for both options and did not strongly influence the decision."

This explanation provides key insights into the model's decision-making. The planner appears to weigh multiple objectives, prioritizing progress and comfort while tolerating a mild increase in lane deviation. Interestingly, it only deprioritized rule-based or safety-related costs in this specific context, possibly because neither trajectory violated those constraints. This highlights how interpretable outputs can help us understand not only what the system did, but what it "cared about" in specific scenarios.

About figure 6, **with all the concepts active**, obtained the following insight: "1. The planned trajectory was chosen over the current one because it has a lower overall cost, particularly in the Cost Volume component (0.2446 for current vs 0.1172 for planned). This suggests that the model's learned preferences for comfort, efficiency, and risk avoidance are better met by the planned trajectory based on training data.

2. The model weighted the Cost Volume more heavily in making this decision, as it represents a learned component that captures latent preferences for comfort, efficiency, and risk avoidance. Lower values of Cost Volume indicate a better outcome in terms of these factors.

3. In this specific case, the safety, headway, lane divider, rule costs were identical or lower for the planned trajectory compared to the current one. This means that the current trajectory did not pose immediate risks or violate any rules, but the Cost Volume weighed more heavily in favor of the planned trajectory due to its learned preferences.

4. The progress cost is a measure of how much progress is being made towards the destination. In this case, lower values indicate better outcomes because more progress has been made. The planned trajectory had a slightly lower progress value (0.9902 vs 0.9927), suggesting that it may lead to faster or more efficient travel, though this is ultimately offset by the Cost Volume's preference for it. Despite the current trajectory exhibiting slightly better progress (0.9927 vs. 0.9902) and incurring no meaningful penalties in safety, headway, rule, or lane divider costs, the planner still favored the alternative route due to its significantly lower Cost Volume (0.1172 vs. 0.2446)."

This indicates that the learned component (Cost Volume) is not simply one factor among many—it often serves as the tie-breaker or even the primary decision-maker when multiple trajectories are similar in terms of rule compliance or physical feasibility.

Moreover, this preference shows that the model has learned nuanced, data-driven notions of what constitutes a "better" trajectory—beyond human-specified costs—likely factoring in comfort, risk buffering, and expected motion behavior patterns.

In short, the model prioritizes trajectories that align with training-derived experience, even at the expense of slightly slower progress, reinforcing the importance of incorporating learned priors into modern planning systems for more human-like and context-aware decision-making.

For figure 7, the LLM, **with no rule concept**, stated:

"1. The planned trajectory was chosen over the current one primarily because it has significantly lower costs in most of the key components: - Safety: The planned trajectory has no safety concerns as its safety cost is zero compared to 0.4007 for the current trajectory, indicating a safer path. - Headway: Similarly, the headway cost for the planned trajectory is zero compared to 0.3035 for the current one, suggesting better longitudinal spacing. - Lane divider: Both trajectories have an equal lane divider cost of 0.2500, so this factor did not influence the decision.

2. However, it is important to note that the planned trajectory did not perform as well as the current one in terms of comfort and progress costs: - Costvolume: The costvolume cost for both trajectories is zero, so this factor had no impact on the decision. - Comfort: The planned trajectory has a slightly higher comfort cost (0.0004) compared to the current one (0.2859), indicating that the current path might offer a more comfortable ride. - Progress: Conversely, the planned trajectory has a lower progress cost of 0.7527 compared to the current one's 0.9425, suggesting that it may take longer for the vehicle to reach its destination using the planned path.

3. In making this decision, the model appears to have heavily weighted safety cost over comfort and progress costs, prioritizing safer driving behavior despite potential trade-offs in terms of ride comfort and time efficiency."

An interesting insight from this analysis is that in the absence of the rule concept, the model still prioritizes safety above all else—even when it leads to suboptimal outcomes in terms of comfort

and efficiency. Despite the planned trajectory being less comfortable and slower (higher progress cost), the model selects it due to its zero safety and headway costs.

This suggests that the model has learned a strong internal bias toward minimizing collision risk—even without explicit rule-based reasoning, indicating that certain safety behaviors can emerge implicitly through cost-based optimization.

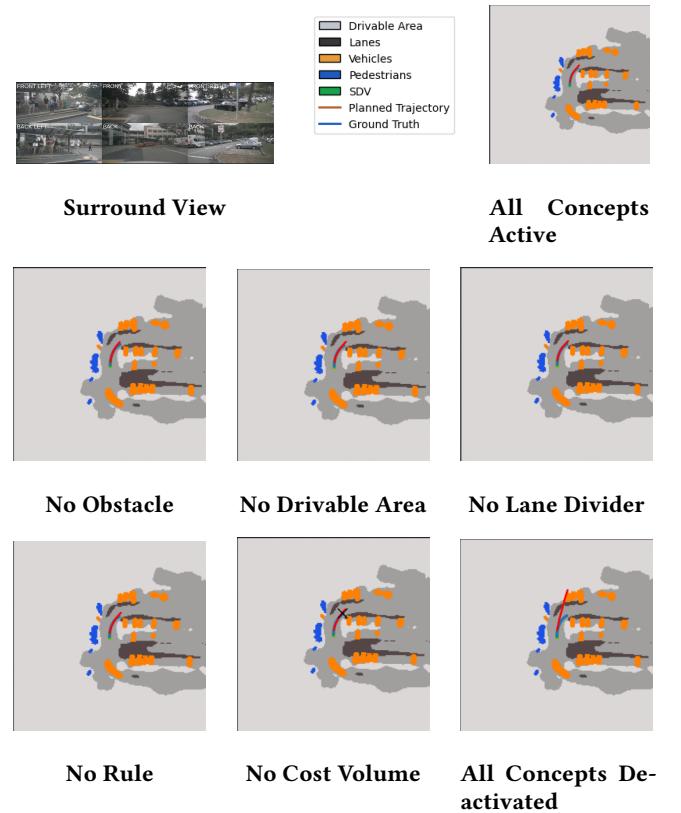


Figure 5: Comparison of planning results under individual concept deactivations (Scenario 09). When a concept is deactivated, its associated cost is ignored during trajectory selection.

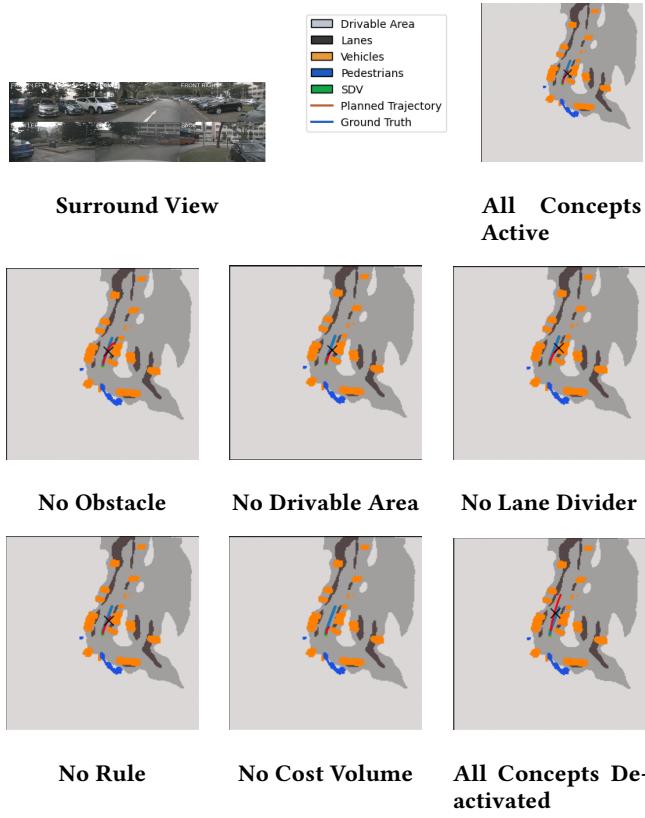


Figure 6: Comparison of planning results under individual concept deactivations (Scenario 16). Each configuration omits a specific semantic cost. The base trajectory remains fixed across all configurations.

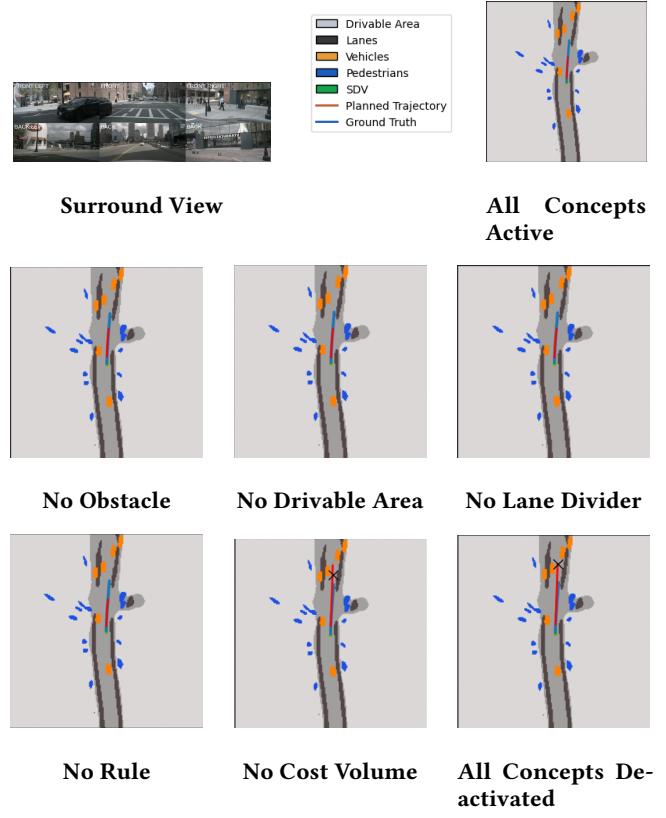


Figure 7: Comparison of planning results under individual concept deactivations (Scenario 36). Each configuration omits a specific semantic cost. The base trajectory remains fixed across all configurations.

.2 Extended "Forgetting to See" Experiments and Analysis

This section presents further experiments exploring the impact of selectively removing semantic concepts from the input representation and analyzing the resulting trajectories.

Collision Visualization. In Figure 8 (Scenario 27), we analyze the impact of removing vehicle perception from the semantic input. When all vehicle concepts are active, the planned trajectory anticipates the curve by adjusting the path to maintain a safe distance from parked vehicles, resulting in a smoother and safer navigation. Conversely, when vehicle perception is removed, the modified trajectory maintains a constant and follows a nearly straight path, failing to steer away from the parked cars. This leads to a less cautious path that could potentially increase the risk of collision or uncomfortable maneuvers near obstacles. The results emphasize the importance of explicit vehicle detection for adaptive trajectory planning, especially in complex road scenarios involving static obstacles.

In Figure 9 (Scenario 49), we observe a scenario where the model fails to perceive the vehicle directly ahead due to the removal of the corresponding semantic concept. When all concepts are active, the planned trajectory safely slows down and avoids collision by

maintaining distance. However, with the frontal vehicle representation removed, the model plans a trajectory that leads to a frontal collision. This highlights the crucial role of explicit obstacle detection in preventing unsafe behaviors and ensuring collision-free navigation.

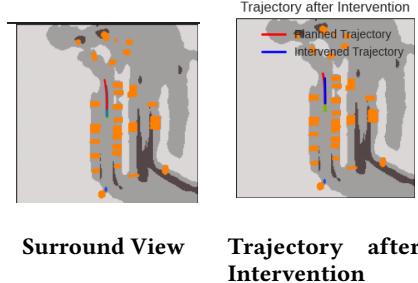


Figure 8: Comparison between the planned trajectory with full vehicle perception (solid line) and the modified trajectory without vehicle perception (dashed line). The full perception steers away from parked vehicles, while the modified path maintains speed and follows a nearly straight line.

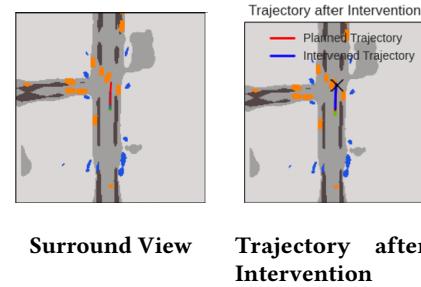


Figure 9: Comparison with full semantic concepts (solid line) versus without frontal vehicle perception (dashed line). The full model slows down and maintains distance to avoid collision, while the modified model fails to detect the vehicle ahead, resulting in a collision course.