Driving through the Concept Gridlock: Unraveling Explainability Bottlenecks in Automated Driving

Jessica Echterhoff[1], An Yan[1], Kyungtae Han[2], Amr Abdelraouf[2], Rohit Gupta[2], Julian McAuley[1]
[1]University of California, San Diego
[2] InfoTech Labs, Toyota Motor North America R&D
[1] {jechterh,ayan,jmcauley}@ucsd.edu [2] {kt.han,amr.abdelraouf,rohit.gupta}@toyota.com

Abstract

Concept bottleneck models have been successfully used for explainable machine learning by encoding information within the model with a set of human-defined concepts. In the context of human-assisted or autonomous driving, explainability models can help user acceptance and understanding of decisions made by the autonomous vehicle, which can be used to rationalize and explain driver or vehicle behavior. We propose a new approach using concept bottlenecks as visual features for control command predictions and explanations of user and vehicle behavior. We learn a human-understandable concept layer that we use to explain sequential driving scenes while learning vehicle control commands. This approach can then be used to determine whether a change in a preferred gap or steering commands from a human (or autonomous vehicle) is led by an external stimulus or change in preferences. We achieve competitive performance to latent visual features while gaining interpretability within our model setup. [1]

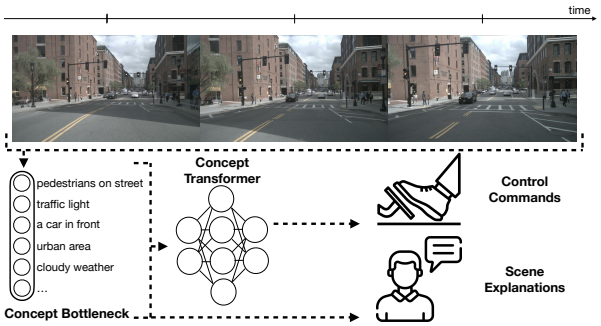

Figure 1. Our proposed framework combines the power of concept bottlenecks and Longformer [4] architecture to enable interpretable prediction of control commands in automated driving. By incorporating human-defined concepts within the concept bottleneck layers, we unravel the explainability bottlenecks for safer and more reliable driving. The Longformer architecture allows capturing long-range sequential dependencies in driving scenarios and reveals interesting subsequences through its attention mechanism, while the concept bottlenecks enhance transparency explaining these through driving-related concepts.

1. Introduction

Understanding how human drivers and autonomous vehicles make decisions is essential to ensure safe and reliable operation in various real-world scenarios. Neural networks are powerful tools used for automated learning in the field of self-driving cars [2, 8, 25, 28, 30, 34, 37]. However, one significant challenge associated with deep neural networks is their nature as black-box models, which hinders the interpretability of their decision-making process. This paper proposes to address this challenge by applying concept bottleneck models for explaining driving scenarios. Concept bottleneck models incorporate vision-based human-defined concepts within a bottleneck in the model architecture [22, 31]. By encoding driving and scenario-related concepts into the decision-making process, our objective is to provide interpretable and explainable insights into the factors that influence the actions of both drivers and autonomous vehicles. Previous research has demonstrated the effectiveness of learning vehicle controls for autonomous driving [6, 24, 39, 42, 42], but the lack interpretability poses challenges to trust, safety, and regulatory compliance. The development of interpretable and explainable models has thus gained significant attention in the research community, aiming to bridge the gap between the performance and interpretability of deep learning models.

Our proposed procedure offers a novel approach to address this interpretability gap in a sequential setup. By incorporating human-defined concepts into the bottleneck of the model architecture, we provide a means to understand and interpret the decision-making process of drivers and autonomous vehicles. Our results can be used for driver inter-

[1]The code for this work is available at https://github.com/jessicamecht/concept_gridlock.

vention prediction in applications such as adaptive cruise control or lane keeping.

This work provides the following contributions:

- We propose a novel pipeline for explainable driving that builds concepts with large language models, converts image features into explicit concept scores, and then learns sequential patterns with a Longformer architecture. We provide extensive experiments around model architectures and feature backbones including traditional approaches such as Residual Neural Networks (ResNet), Contrastive Language-Image Pretraining (CLIP) models, and Vision Transformers (ViT) for both single and multi-task setups.
- We find that the concept space maps accurately to different driving conditions and that we can use our transformer attention mechanism to select when to reveal automated system explanations to a driver, highlighting the utility of concept bottleneck models for the rather unexplored sequential settings.

Our experimental results demonstrate the effectiveness of concept bottleneck models in sequential learning. Our interpretability analysis reveals that our concept bottleneck models offer insights into the factors influencing a driver's and subsequently a model's decision-making process, enhancing transparency and trustworthiness in autonomous driving systems. For example, we show that they can explain changes in driving behavior such as change of forward distance and give reasoning for those changes.

2. Related Work

2.1. End-to-End Learning of Vehicle Controls

Research in automated driving has examined perception-based tasks such as finding lane markings, traffic lights, recognizing traffic participants [6, 24, 39] as well as end-to-end processes to learn vehicle controls [5, 42]. For example, Xu et al. [42] explore a stateful model using a dilated deep neural network and recurrent neural network to predict future vehicle motion given input images. Bojarski et al. [5] train a deep neural network to map front-facing video frames to steering controls. Hecker et al. [12] explore an extension of a model taking multiple modalities as input for control prediction. Different approaches use behavioral cloning to learn a driving policy as a supervised learning problem over observation-action pairs from human driving demonstrations [24], but only a few explain the rationale for system decisions [21], which makes their behavior opaque and un-interpretable.

2.2. Concept Bottleneck Models

Using human concepts to interpret model behavior has been drawing increasing interest [3, 18]. Concept bottleneck models [22] extend the idea of first predicting image concepts, then using these concepts to predict a classification target [23]. Original concept bottleneck models learn the concept space jointly or sequentially with a classification or regression task [22]. These models introduce interpretability benefits, but require training the model using concept and class labels, which can be a key limitation. Label-free concept bottleneck models [31] or models with unsupervised concepts [35] alleviate this problem. Most of the work on concept bottleneck models evaluates supervised classification task setups [22, 31, 35, 43, 44].

The evaluation of concept bottleneck models in sequential settings remains relatively unexplored. Notably, concept bottleneck models enable the identification of key factors or features that can contribute to driving decisions. Extracted concept representations from input data can highlight relevant information that is driving the predictions. Sequential evaluations provide valuable insights while capturing temporal dependencies and understanding how concepts evolve over time in dynamic scenarios such as driving. Our work focuses on evaluating concept bottleneck models in sequential tasks to assess their performance and interpretability in dynamic decision-making domains.

2.3. Vehicle Action Explanations

The importance of explanations for an end-user has been studied from the psychological perspective [26, 27] indicating the benefit of explanations in autonomous driving. Different work focuses on visual explanations [14, 15, 20]; *e.g.* Wang et al. [41] introduce an instance-level attention model that finds objects that the network needs to pay attention to. Such visual attention might not be convenient for users to "replay" (in the driving domain). It is therefore important to be able to justify the decisions made and explain why they are reasonable in a convenient manner, *e.g.* in natural language [14, 15, 20]. Previous research in the field of explainable decision-making in autonomous vehicles explores the use of recurrent neural networks for explanation generation. Kim et al. [21] use an architecture based on a convolutional image feature encoder and learn vehicle sensor measurements such as speed while aligning temporal and spatial attention. Their explanation generation process uses an LSTM [17] to predict next-word probabilities. In contrast, our work demonstrates the potential of concept bottleneck models in providing insights into the decision-making process. To incorporate scene information, Kim et al. [19] use an active approach to feed human-to-vehicle advice into the vehicle controller. However, this requires *a priori* information from the human on a situation that is often difficult to obtain. Similarly, Kim et al. [20] propose a system to learn vehicle control with the help of human advice. Those works show that human advice is useful, but do not directly explain why a particular model makes a particular decision.

Despite these advancements, there is still a need for further research to develop robust and effective approaches for explaining driver and autonomous vehicle decisions. Existing studies focus on specific aspects of *post-hoc* explanations or how to use explanations *a priori*, but a framework that integrates human-defined concepts for automated driving *in-situ* within the model to enable white-box model explanations is lacking. Our paper addresses this gap by proposing a novel approach that utilizes concept bottleneck models to encode various driving-related concepts within the decision-making process. By incorporating concepts we aim to provide a holistic understanding of the factors influencing driver and autonomous vehicle actions from within the model.

3. Methods

Consider predicting a target value $y \in \mathbb{R}$ from input $x \in \mathbb{R}^d$, while trying to gain reasoning c for the prediction of the target value. That is, we observe training points $\{(x^{(i)}, y^{(i)})\}_{i=1}^n$, and we want to determine $y^{(i)}, c^{(i)}$ where $c^{(i)} \in \mathbb{R}^k$ is a vector of k concepts. We consider bottleneck models of the form $f(g(x))$, where $g : \mathbb{R}^d \to \mathbb{R}^k$ maps an input x into the concept space ("clear skies", "a car in the lane ahead in close proximity", etc.), and $f : \mathbb{R}^k \to \mathbb{R}$ maps concepts into a final prediction (*e.g.* forward distance is 40 meters). These types of models are called concept bottleneck models [22, 31] because the control command prediction $\hat{y} = f(g(x))$ relies on the input x through the bottleneck prediction $\hat{c} = g(x)$.

3.1. Image Feature Backbone

Our method takes inspiration from video vision transformer networks [1, 29]. Typically, spatial backbones take on the function of $g : \mathbb{R}^d \to \mathbb{R}^k$ that maps an input x into the latent (un-interpretable) feature space, (*e.g.* Neimark et al. [29] use the video vision transformer from Arnab et al. [1] as a latent feature bottleneck). However, we incorporate explainability through concept bottleneck models [22]. This pre-trained conceptual spatial backbone operates as a learned feature extraction module to determine sequential decisions or control commands. We compare this model with traditional convolutional- or transformer-based methods [1, 9–11, 45]. Let these features F_{input} denote the input feature to the subsequent sequential evaluation component (Longformer).

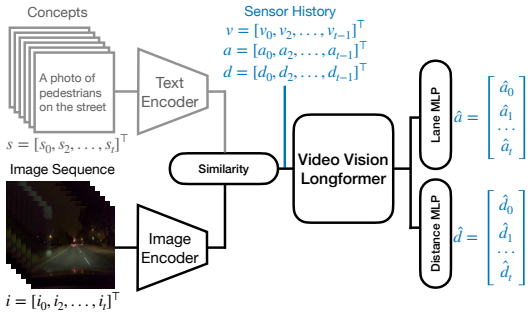

Figure 2. Pipeline of the interpretable concept bottleneck control command prediction. v, a and d denote the sensor history of speed, steering angle and leading vehicle distance.

3.2. Concept Bottleneck

When replacing the feature backbone with a concept bottleneck model we construct driving scenarios s. These scenarios are supposed to describe scenes and encode contextual information about the driving scenario in natural language. A scenario captures factors such as road conditions, traffic density, and weather conditions. To obtain those scenarios, we leverage two concept curation methods. First, we use the generative capabilities of GPT-3.5 [32] to create diverse driving scenarios. We specifically ask the language model to provide scenarios as described in the following, starting with very general scenarios and subsequently generating more fine-grained scene explanations.

- List scenarios that could occur in traffic starting each sentence with $\{$*a photo of ...*$\}$
- List scenarios that could occur in traffic with respect to $\{$*weather; traffic participants, lane changing, highway driving, city driving*$\}$ starting each sentence with $\{$*a photo of ...*$\}$

Like Radford et al. [33], we follow the template of $\{$*a photo of ...*$\}$ (*e.g. a photo of a car driving on a highway*) as a default, as it has been shown that the performance of specific concept bottleneck models can be increased this way [33].

These generated scenarios are then combined with a subset of existing human-created scene descriptions from the NuScenes dataset [7]. We transform these descriptions into the same pattern from Radford et al. [33]. This allows us to enrich the dataset with other diverse driving contexts, *e.g.* "pedestrians" or "workers on the street", as well as compare different concept curation methods. We then manually filter the set for obvious duplicates. The specific construction of the concept space $\mathbb{S}$ is domain-specific and can be customized for other driving-related domains. To the best of our knowledge, this is the first captured driving-related concept bottleneck, and we release these scenarios and code upon publication.

In the concept bottleneck model g, we encode the image features x using an image encoder $g_{\text{image}} : \mathbb{R}^d \to \mathbb{R}^l$ and scenarios $s \in \mathbb{S}$ using a text encoder $g_{\text{text}} : \mathbb{R}^s \to \mathbb{R}^l$ [33]. For each image, we can then measure the similarity between the embedding $g_{\text{image}}(x)$ and the scenarios $g_{\text{text}}(s)$ employ-

ing cosine similarity:

$$\text{sim}_{\cos}(x, s) = \frac{g_{\text{image}}(x) \cdot g_{\text{text}}(s)}{\|g_{\text{image}}(x)\| \|g_{\text{text}}(s)\|} \quad (1)$$

where $\cdot$ represents the dot product, and $|\cdot|$ denotes the Euclidean norm to get an indication of what is happening in image frames from the driving sequence.

3.3. Temporal Encoder

Video vision transformers encode visual features in a temporal manner with a transformer architecture that was originally developed for natural language processing [40]. This acts as our regression module $f : \mathbb{R}^l \rightarrow \mathbb{R}$ for each frame encoded with g. The attention mechanism in a transformer neural network is given by

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (2)$$

where where Q, K, and V are the query, key, and value matrices, respectively, and d is the dimensionality of the key vectors. The softmax function normalizes the attention weights. Due to the original transformer attention complexity of $O(n^2)$, a Longformer architecture with a sliding window attention [4] is useful to reduce computational overhead [29]. Given a sliding window size w and sequence length n, its complexity is reduced to $O(w \times n)$ [4]. To attend to other time steps in the sequence, we use a window attention size of eight frames. The sequence of feature vectors from the backbone and the sensor history of previous vehicle speed v, steering angles a and distance to leading vehicles d for each captured frame is fed to the Longformer model as shown in Fig. 2. We prepend a special token ([CLS]) at the beginning of the feature sequence. The Longformer maintains global attention on that special [CLS] token. After propagating the sequence through the Longformer layers, we use the final state of the features related to this classification token as the final representation of the video and apply it to the given regression task head to learn the control commands through a linear regression head. Each output from the temporal encoder is processed with a Multi-Layer Perceptron (MLP) head to provide a final predicted value. The MLP head contains two linear layers with a GELU [16] non-linearity and dropout [38] between them. The input token representation is processed with layer normalization. We use one MLP for each task trained separately, and two MLPs for the multi-task setup. The idea for evaluating multi-task setups is that a person might be more careful in their overall driving behavior for both tasks, so that the two tasks could benefit from being trained together. We train our models with Root Mean Squared Error (RMSE) loss $\mathcal{L} = \sqrt{\frac{\sum_{i=1}^{N}(f(g(x))-y)^2}{N}}$ with $g(x) = sim(x, s)$.

4. Experiments

4.1. Data

For comparative evaluation, we employ two datasets consisting of diverse driving scenarios captured from real-world driving situations. The datasets encompass a wide range of environmental conditions, traffic scenarios, and driver behaviors to ensure generalizability of our findings.

Comma2k19. We explore the Comma 2k19 dataset [36], which captures commute scenarios with different features, *e.g.* visual images, CAN data (*e.g.* steering wheel angle), and radar data (distance to preceding vehicle) in the San Francisco Bay Area. The Comma data mostly consists of highway scenarios, and their captured sequences are comparatively long compared to other datasets. In total, Comma 2k19 has 100GB of data from 33 hours of driving. In this work, we use a 25GB subset of the data. The data was captured at 20fps and was subsampled to 4fps to reduce redundancy for training. All data sequences are one minute long, but continuous driving sequences per session ranged between 3 and 13 minutes. For our purposes, each driving sequence consists of 240 samples.

NuScenes. The NuScenes dataset [7] is collected using a fleet of autonomous vehicles equipped with lidar, radar, cameras, and ego-motion sensors, and is designed for the development and evaluation of perception, planning, and control algorithms. With data captured in various urban driving scenarios across multiple cities, the NuScenes dataset provides researchers and developers with a range of environments and traffic conditions to analyze. The dataset includes annotations for each sensor modality, including 3D bounding boxes, as well as natural language scene descriptions, enabling algorithm development and evaluation in a structured manner. Each scene of the dataset consists of 20 seconds and is resampled at 1fps. The descriptions serve as ground truth for our the concept bottlenecks. We use a subset of 250 scenes for evaluation of our method.

The two datasets serve different purposes. (1) the comma dataset provides long driving sequences to learn potential interventions on highway scenarios, that can be connected to explanatory driving behavior. For example, we might like to explain a change in leading-vehicle gap, occurring due to a change of scenario (*e.g.* someone cut in the front lane), versus changed user preferences. (2) the NuScenes dataset with its natural language scene annotations can evaluate the explanatory abilities of our model. These two datasets capture a wide variety of scenarios in city and highway driving. For both datasets we resize all frames to 224×224 pixels. We exclude distances over 70m for our evaluation, as we empirically evaluated that distances beyond this threshold contain little visual information useful for gap prediction (*e.g.* no leading vehicle present). We use a 0.85/0.05/0.1 train/val/test split.

Dataset	Model	Feat. Size	a-MAE	d-MAE	(a,d)-MAE
Comma	ResNet+GapFormer [34]	512	0.08	0.28	-
Comma	CLIP+Longformer	512	0.03	7.95	[0.22, 8.97]
Comma	ViT+Longformer	768	0.06	5.23	[0.8, 6.08]
Comma	ResNet+Longformer	512	0.03	3.79	[0.37, 4.11]
Comma	Concept (Full)+Longformer	643	0.7	0.97	[0.36, 1.83]
Comma	ResNet+Concept (Full)+Longformer	1,155	0.37	2.43	[2.15, 1.74]
NuScenes	ResNet+GapFormer [34]	512	0.57	0.74	-
NuScenes	CLIP+Longformer	512	0.57	5.46	[3.51, 3.47]
NuScenes	ViT+Longformer	768	3.75	1.31	[0.44, 16.62]
NuScenes	ResNet+Longformer	512	5.87	26.5	[9.47, 43.81]
NuScenes	Concept (Full)+Longformer	643	1.89	4.21	[0.36, 6.65]
NuScenes	ResNet+Concept (Full)+Longformer	1,155	0.97	4.8	[2.46, 4.26]

Table 1. Mean Absolute Error (MAE) performance of different models on the downstream task of steering angle (a) and distance (d) prediction in a single and multi-task setting, compared to the inherently explainable concept bottleneck model.

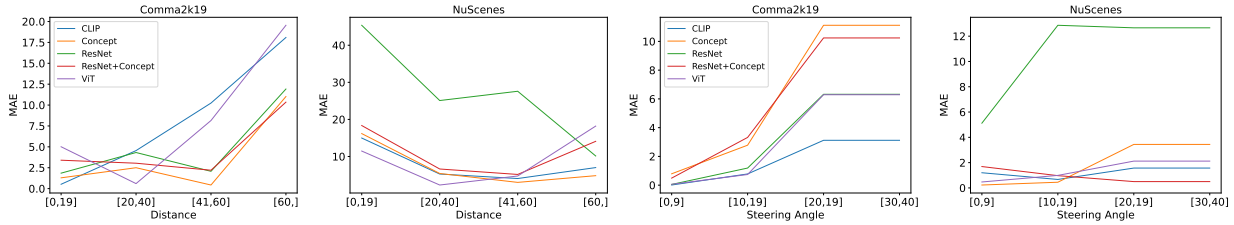

Figure 3. Error analysis of different model backbones and tasks. We see fewer absolute error on smaller ground truth forward distances and steering angles. This intuitively makes sense as the visual information is clearer for small gaps (e.g. when directly following a leading vehicle), compared to longer distances. Similarly, learning small steering angles is easier, *e.g.* for lane keeping in highway driving, compared to turning on an intersection. Our concept bottleneck model continually performs similarly or better to other approaches.

4.2. Backbones

To identify how explainable concept bottleneck models perform compared to standard methods, we conduct an analysis of different backbone models. We evaluate the performance of ResNet-18 [10], Vision Transformer [1], and CLIP [33] backbones for our single- and multi-task control command prediction task. ResNet-18 [10], with its deep architecture and skip connections, has been a benchmark backbone model in computer vision. Vision Transformer (ViT) [1] replace convolutional layers with self-attention mechanisms, with an ability to capture global dependencies. We investigate the performance of the CLIP image backbone, which is another transfomer-based backbone, but typically its ViT-based image encoder is combined with a transformer-based text encoder. We analyze its effectiveness in capturing visual-semantic representations with only its image-based encoder. Concept bottlenecks can provide additional linguistic explainability without requiring additional language generation models (such as LSTMs in [21]).

5. Results

5.1. Control-Command Prediction Performance

We evaluate the performance of concept bottleneck models as interpretable feature extractors for downstream tasks. Tab. 1 presents the Mean Absolute Error (MAE) performance of different black-box backbones compared to the concept bottleneck model on the tasks of steering angle and distance prediction in both single and multi-task settings. It can be observed that the concept bottleneck models achieve a competitive MAE across different datasets. In particular, for the Comma dataset, the concept bottleneck model with a feature size of 643 obtains with a MAE of 0.7 for angle prediction and 0.97 for distance prediction. Similarly, for the NuScenes dataset, the concept bottleneck model with a feature size of 643 achieves a MAE of 1.89 for angle prediction and 4.21 for distance prediction. These results indicate that concept bottleneck models exhibit good performance as interpretable feature extractors for downstream tasks. We see no significant difference in performance of our concept bottleneck approach between single- and multi-task setups. In Fig. 3, we show error based on different ground-truth magnitudes. We observe that concept bottleneck models as feature extractors can lead to better performance of control command prediction, while convolution-based approaches may fail to learn the task (on the NuScenes dataset). However, when the visual properties are connected more strongly to the task (e.g. for gap prediction, compared to steering angle prediction), we see an increased utility and performance. In terms of computa-

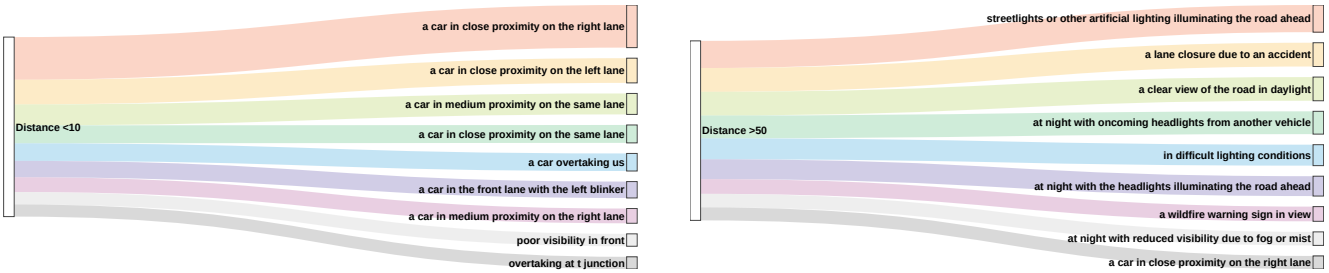

Figure 4. Visualization of explanation capabilities of our model to determine reasons drivers keep short forward distances (*e.g.* distance < 10 meter) or longer distances (*e.g.* distance > 50 meter). Height of the lines indicates fraction of top-10 predictions.

Concept Curation	Comma2k19	NuScenes
Human	1.77	3.93
GPT-3.5	0.89	2.02

Table 2. Comparison of concepts that were created by humans (adapted from [7]) versus curated from GPT-3.5 [32] for predicting lead vehicle distance. We can see that automatically curated concepts can perform better in terms of distance MAE compared to human curated concepts.

tion speed, our prediction procedure has an average model inference latency over 100 runs of 0.1 seconds (excluding data processing), and system throughput (including data processing) of 1 per second using an NVIDIA RTX A6000 GPU for long sequences (240 frames on Comma2k19) and 2 per second for short sequences (20 frames on NuScenes).

5.2. Scene Explanation Capabilities

By employing the concept bottleneck model, we can analyze and interpret the factors contributing to larger or smaller gaps to leading vehicles. The interpretability of the concept bottleneck model allows us to understand the underlying causes behind these gap variations, shedding light on human decision-making processes in relation to preceding vehicles. In Fig. 4, we see that smaller gaps are typically associated with a prediction of *"vehicles in close and medium proximity"* or *"cars in the front lane"*. On the other hand, large distances are associated with the prediction of *"a clear view"*, *"difficult lighting conditions"*, or *"at night"*. Intuitively a driver might keep a larger leading distance at night, and shorter distances in *e.g.* a traffic jam.

To quantitatively evaluate the effectiveness of explainability through the concept bottleneck, we design a human evaluation study of 50 images per dataset. We evaluate the top-10 concepts for each frame and extract the top-3 occurring concepts over 20 frames. We present each short video with the predicted concepts to three human crowdworkers and ask them how many of the concepts are correct. When we aggregate the worker votes by majority vote, we find that 94% of the top-3 concept predictions have at least one correct concept for NuScenes and 90% for Comma2k19. A fine-grained evaluation of individual reviews (not majority voted) shows that (for NuScenes/Comma) 9%/15% of instances are labeled as having no correct concepts, 30%/32% as having one correct concept, 38%/34% have two correct concepts and 23%/19% have all top-3 concepts correct.

Additionally we calculate the common content words between NuScenes scene descriptions and concept predictions, such that the NuScenes descriptions serve as a form of ground-truth. We consider the top-3 concept predictions for each frame and then the top-3 concept predictions for the entire scene, and remove any stopwords to only evaluate relevant content words for each scene and scenario from the concept bottleneck. By considering the top-3 predicted concepts, we are able to correctly explain 81% of the scenes. When considering the top-1 concept prediction, we can explain 76% of all scenes accurately. This demonstrates the capability of our approach to effectively explain the content of scenes by leveraging concept predictions and their intersection with scene descriptions. In Fig. 5, we also show predictions of the concept model on driving scenarios.

5.3. Concept Curation

Concept curation plays a vital role in building a comprehensive understanding of the automated driving domain. Traditionally, it has relied on human experts who bring their expertise and domain knowledge to the curation process. Humans can provide nuanced insights, contextual understanding, and connections between different concepts based on their experience, but they are also costly and subjective. Human curation can be time-consuming, limited by individual biases, and susceptible to errors or omissions. We evaluate a randomly selected subset of 270 human created concepts, adapted with the template from Sec. 3.2 from the scene descriptions of the NuScenes dataset. These textual descriptions were made by expert annotators to add captions for each scene (e.g.: "Wait at intersection, peds on sidewalk, bicycle crossing, jaywalker") [7]. We additionally evaluate 270 generated concepts by GPT 3.5 similar to [32], yielding sentences like "driving on a highway with an overpass

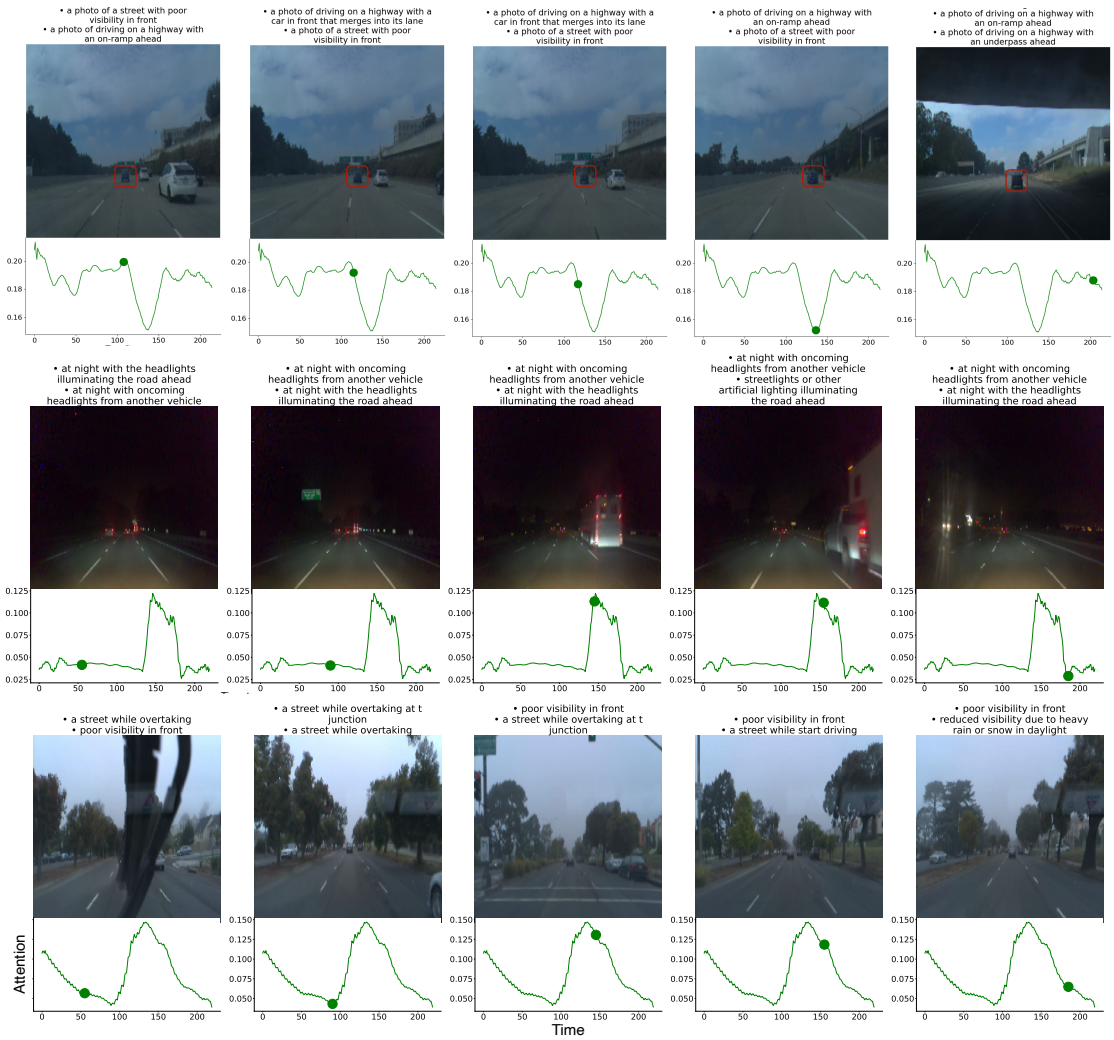

Figure 5. Three scenarios from the Comma dataset for gap prediction with scenario explanations from the concept bottleneck and attention values (y-axis) over time (x-axis) at particular points in time (green dot). We observe that the Longformer attention is a good indicator for when interventions might happen. For example, we see a leading car changing the lane, leading to attention drop (top) or a scenario of passing at a T-junction, leading to a spike (bottom); or a scenario where the ego vehicle passes a trailer vehicle on the right, leading to an attention spike (middle).

overhead". Our results show that human curation is not better compared to concept curation by large language models (Tab. 2). On the Comma2k19 dataset we achieve a distance MAE of 0.89 for GPT curated concepts, and 1.77 for human curated concepts on the Comma dataset, and MAE of 2.02 for GPT curated concepts, and 3.93 for human curated concepts on the NuScenes data Tab. 2.

5.4. Does attention matter?

We provide an analysis of three scenarios extracted from the Comma2k19 dataset, focusing on the gap prediction task (Fig. 5). These scenarios are accompanied by explanations generated from the concept bottleneck model, offering insights into the underlying factors influencing the observed gap variations. In our analysis, we investigate the role of the Longformer attention mechanism as a valuable indicator for identifying instances where interventions might occur. By examining the attention, we can discern patterns and changes in the scene that may prompt user intervention. In the first scenario, we observe the ego car changing lanes. This maneuver often requires careful monitoring and potential intervention from the driver. By examining the attention distribution captured by the Longformer, we notice a significant drop in attention at the moment when the ego car changes the lane and has a free lane ahead, and an attention increase when it is back in a lane behind a vehicle. This attention drop suggests that the concept bottleneck model correctly identifies this critical event and recog-

Dataset	Feat. Size	a-MAE	d-MAE
Comma	24	0.38	3.48
Comma	48	0.3	1.15
Comma	100	**0.27**	**0.22**
Comma	300	0.43	0.6
Comma	Full	0.7	0.97
NuScenes	24	1.49	1.85
NuScenes	48	0.60	**0.02**
NuScenes	100	**0.52**	0.51
NuScenes	300	2.27	4.01
NuScenes	Full	1.89	4.21

Table 3. Bottleneck size (randomly selected from full (643) bottleneck) versus command control prediction performance. We observe that bottleneck size seems to have a significant impact on performance, with a sweet spot at 100 concepts.

nizes the reduced relevance of certain features in the scene. In the second scenario, we encounter a situation where the ego vehicle passes a trailer vehicle on the right side. This scenario often demands extra caution and anticipation from the driver, as the presence of large vehicles can impact the driving environment. Analyzing the attention distribution, we observe a spike in attention when the trailer vehicle enters the scene. This attention spike indicates that the model successfully captures the significance of this change in the scenery, identifying the trailer vehicle as a prominent object that requires increased attention. The third scenario involves a noteworthy change in the driving environment on a T-junction while driving in the rain. As we analyze the attention patterns, a spike in attention occurs when our vehicle encounters the junction situation and the attention serves as an indicator for the change in the scenery. The model, through its attention mechanism, effectively recognizes and highlights these critical moments, and can explain these scenarios through its bottleneck activations.

We evaluate the Longformer attention to observe whether it can be used to select when to reveal a concept to a user. If a particular part of a sequence in (semi-) autonomous driving is of relevance, indicated by the attention, it can be used to decide if an intervention from the autonomous car is required, if the user should take over and provide reasond why (given the concept explanation).

5.5. Does Bottleneck Size Matter?

We investigate the impact of bottleneck size on the performance, to evaluate how the size of the bottleneck affects the accuracy of control command predictions. The ablation study varies the size of the bottleneck while keeping all other factors constant. The ablation study results are summarized in Tab. 3. The feature size denotes the number of concepts in the bottleneck layer, which were randomly drawn from all possible 643 scenarios. We observe that as the bottleneck size increases, both steering angle MAE and distance MAE decrease. For Comma data with a bottleneck size of 24, the steering angle MAE is 0.38 and distance MAE is 3.48. With a bottleneck size of 100, the steering angle MAE decreases to 0.27, and the distance MAE drops to 0.22. Interestingly, further increasing the bottleneck size to 300 resulted in worse performance. We observe a similar tendency for the NuScenes dataset: increasing the bottleneck size leads to improved prediction performance up to a certain threshold after which we observe decreasing performance. With a bottleneck size of 24, the steering angle MAE is 1.49 and distance MAE is 1.85. Increasing the bottleneck size to 100 reduces both steering angle MAE (0.52) and distance MAE (0.51) with performance degradation for larger concepts. The findings indicate an impact of bottleneck size on the prediction accuracy, with a "sweet spot" at a bottleneck size of 100 concepts. There are different reasons for the performance benefits with smaller concept sizes. Previous work shows that it is possible to achieve good performance with smaller concept spaces [44] and that correlated concept spaces can be an issue [13]. We conjecture that the performance benefit in our work for a smaller concept space may be based on (1) multiple concepts in the original concept set having only small deviations, which means that we might achieve the same or better results when excluding them. For example, the difference between the concept "a pedestrian crossing"; "a pedestrian crossing crosswalk"; "a pedestrian crossing traffic light" can be subtle. (2) the image size of 224×224 pixels does not allow for fine-grained concept granularity. For example different street signs like "a curve sign"; "a steep hill sign"; "a winding road sign" can be too fine-grained to be visible.

6. Conclusion

This study validates the effectiveness of concept bottleneck models for explainability in sequential settings for automated driving. Our work leverages a concept bottleneck model and Longformer sequential processing unit within a control command prediction setup and we show competitive performance to standard black-box approaches. Using our method, we identify and explain factors contributing to changes in driving behavior both visually through linguistic explanation as well as temporally through transformer attention. This can explain *e.g.* changes in forward distance to a leading vehicle, and enable a deeper understanding of the decision-making processes in automated driving. Our model demonstrates effectiveness in explaining scene content, which can serve as a baseline for future work aligning linguistic, visual and temporal explanations. Future work could explore more use-cases (such as speed prediction), fuse more modalities into the prediction procedure, or analyse bottleneck uncertainty (e.g. with test-time interventions) in more detail.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. Vivit: A video vision transformer. *CoRR*, abs/2103.15691, 2021. 3, 5

[2] Mohammadhossein Bahari, Saeed Saadatnejad, Ahmad Rahimi, Mohammad Shaverdikondori, Amir Hossein Shahidzadeh, Seyed-Mohsen Moosavi-Dezfooli, and Alexandre Alahi. Vehicle trajectory prediction works, but not everywhere. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17123–17133, June 2022. 1

[3] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3319–3327, 2017. 2

[4] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. 1, 4

[5] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016. 2

[6] Martin Buehler, Karl Iagnemma, and Sanjiv Singh. *The DARPA urban challenge: autonomous vehicles in city traffic*, volume 56. springer, 2009. 1, 2

[7] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 3, 4, 6

[8] Gowtham Garimella, Joseph Funke, Chuang Wang, and Marin Kobilarov. Neural network modeling for steering control of an autonomous vehicle. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2609–2615. IEEE, 2017. 1

[9] Simon Ging, Mohammadreza Zolfaghari, Hamed Pirsiavash, and Thomas Brox. Coot: Cooperative hierarchical transformer for video-text representation learning. *Advances in neural information processing systems*, 33:22605–22618, 2020. 3

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 3, 5

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 3

[12] Simon Hecker, Dengxin Dai, and Luc Van Gool. End-to-end learning of driving models with surround-view cameras and route planners. In *Proceedings of the european conference on computer vision (eccv)*, pages 435–453, 2018. 2

[13] Lena Heidemann, Maureen Monnet, and Karsten Roscher. Concept correlation and its effects on concept-based models. In *WACV*, pages 4780–4788, 2023. 8

[14] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 3–19. Springer, 2016. 2

[15] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Grounding visual explanations. In *Proceedings of the European conference on computer vision (ECCV)*, pages 264–279, 2018. 2

[16] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415, 2016. 4

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, nov 1997. 2

[18] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie J. Cai, James Wexler, Fernanda B. Viégas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International Conference on Machine Learning*, 2017. 2

[19] Jinkyu Kim, Teruhisa Misu, Yi-Ting Chen, Ashish Tawari, and John Canny. Grounding human-to-vehicle advice for self-driving vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10591–10599, 2019. 2

[20] Jinkyu Kim, Suhong Moon, Anna Rohrbach, Trevor Darrell, and John Canny. Advisable learning for self-driving vehicles by internalizing observation-to-action rules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9661–9670, 2020. 2

[21] Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. Textual explanations for self-driving vehicles. In *Proceedings of the European conference on computer vision (ECCV)*, pages 563–578, 2018. 2, 5

[22] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR, 2020. 1, 2, 3

[23] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958, 2009. 2

[24] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, et al. Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE intelligent vehicles symposium (IV)*, pages 163–168. IEEE, 2011. 1, 2

[25] Jinlong Li, Runsheng Xu, Jin Ma, Qin Zou, Jiaqi Ma, and Hongkai Yu. Domain adaptive object detection for autonomous driving under foggy weather. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 612–622, January 2023. 1

[26] Tania Lombrozo. The structure and function of explanations. *Trends in cognitive sciences*, 10(10):464–470, 2006. 2

[27] Tania Lombrozo. Explanation and abductive inference. *Oxford University Press*, 2012. 2

[28] Srikanth Malla, Chiho Choi, Isht Dwivedi, Joon Hee Choi, and Jiachen Li. Drama: Joint risk localization and captioning in driving. In *Proceedings of the IEEE/CVF Winter Con-*

ference on Applications of Computer Vision (WACV), pages 1043–1052, January 2023. 1

[29] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3163–3172, 2021. 3, 4

[30] Chihiro Noguchi and Toshihiro Tanizawa. Ego-vehicle action recognition based on semi-supervised contrastive learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5988–5998, January 2023. 1

[31] Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. *arXiv preprint arXiv:2304.06129*, 2023. 1, 2, 3

[32] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022. 3, 6

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3, 5

[34] Noveen Sachdeva, Ziran Wang, Kyungtae Han, Rohit Gupta, and Julian McAuley. Fast autoregressive transformers meet rnns for personalized adaptive cruise control. In *IEEE International Conference on Intelligent Transportation Systems*, 2022. 1, 5

[35] Yoshihide Sawada and Keigo Nakamura. Concept bottleneck model with additional unsupervised concepts. *IEEE Access*, 10:41758–41765, 2022. 2

[36] Harald Schafer, Eder Santana, Andrew Haden, and Riccardo Biasini. A commute in data: The comma2k19 dataset. *CoRR*, abs/1812.05752, 2018. 4

[37] Nathan A Spielberg, Matthew Brown, Nitin R Kapania, John C Kegelman, and J Christian Gerdes. Neural network vehicle models for high-performance automated driving. *Science robotics*, 4(28):eaaw1975, 2019. 1

[38] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 4

[39] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, MN Clark, John Dolan, Dave Duggins, Tugrul Galatali, Chris Geyer, et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of field Robotics*, 25(8):425–466, 2008. 1, 2

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[41] Dequan Wang, Coline Devin, Qi-Zhi Cai, Fisher Yu, and Trevor Darrell. Deep object-centric policies for autonomous driving. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8853–8859. IEEE, 2019. 2

[42] Huazhe Xu, Yang Gao, Fisher Yu, and Trevor Darrell. End-to-end learning of driving models from large-scale video datasets. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2174–2182, 2017. 1, 2

[43] An Yan, Yu Wang, Petros Karypis, Zexue He, Chengyu Dong, Zihan Wang, Yiwu Zhong, Jingbo Shang, Amilcare Gentili, Chun-Nan Hsu, et al. Robust and interpretable medical image classifiers via concept bottleneck models. *arXiv preprint arXiv:2310.03182*, 2023. 2

[44] An Yan, Yu Wang, Yiwu Zhong, Chengyu Dong, Zexue He, Yujie Lu, William Wang, Jingbo Shang, and Julian McAuley. Learning concise and descriptive attributes for visual recognition, 2023. 2, 8

[45] Jiewen Yang, Xingbo Dong, Liujun Liu, Chao Zhang, Jiajun Shen, and Dahai Yu. Recurring the transformer for video action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14063–14073, 2022. 3