SGDCL: Semantic-Guided Dynamic Correlation Learning for Explainable Autonomous Driving

Chengtai Cao\textsuperscript{1,2}, Xinhong Chen\textsuperscript{1,2}, Jianping Wang\textsuperscript{1,2}, Qun Song\textsuperscript{3}, Rui Tan\textsuperscript{4} and Yung-Hui Li\textsuperscript{5}

\textsuperscript{1}City University of Hong Kong
\textsuperscript{2}City University of Hong Kong Shenzhen Futian Research Institute
\textsuperscript{3}Delft University of Technology
\textsuperscript{4}Nanyang Technological University
\textsuperscript{5}Foxconn Research

\{chengtcao2-c, xinhchen2-c\}@my.cityu.edu.hk, jianwang@cityu.edu.hk, q.song-1@tudelft.nl, tanrui@ntu.edu.sg, yunghui.li@foxconn.com

Abstract

By learning expressive representations, deep learning (DL) has revolutionized autonomous driving (AD). Despite significant advancements, the inherent opacity of DL models engenders public distrust, impeding their widespread adoption. For explainable autonomous driving, current studies primarily concentrate on extracting features from input scenes to predict driving actions and their corresponding explanations. However, these methods underutilize semantics and correlation information within actions and explanations (collectively called categories in this work), leading to suboptimal performance. To address this issue, we propose Semantic-Guided Dynamic Correlation Learning (SGDCL), a novel approach that effectively exploits semantic richness and dynamic interactions intrinsic to categories. SGDCL employs a semantic-guided learning module to obtain category-specific representations and a dynamic correlation learning module to adaptively capture intricate correlations among categories. Additionally, we introduce an innovative loss term to leverage fine-grained co-occurrence statistics of categories for refined regularization. We extensively evaluate SGDCL on two well-established benchmarks, demonstrating its superiority over seven state-of-the-art baselines and a large vision-language model. SGDCL significantly promotes explainable autonomous driving with up to 15.3% performance improvement and interpretable attention scores, bolstering public trust in AD.

1 Introduction

The field of autonomous driving (AD) has witnessed significant strides, mainly owing to recent advancements in deep learning (DL). Despite their efficiency, DL models typically operate as opaque black-box neural networks, offering limited explainability. The significance of explainability in AD is emphasized by various studies that illustrate its influence on public trust and regulatory oversight [Atakishiyev et al., 2021; Omeiza et al., 2021; Goldman and Bustin, 2022; Madhav and Tyagi, 2022; Zablocki et al., 2022]. For instance, Madhav et al. emphasize that increased transparency in the AD’s decision-making processes is crucial for users to trust these systems [Madhav and Tyagi, 2022]. Similarly, the survey conducted by Omeiza et al. finds that the societal acceptance of autonomous vehicles largely hinges on their explainability and trustworthiness [Omeiza et al., 2021].

In explainable autonomous driving (EAD), Xu et al. introduce a new multi-task and multi-label classification paradigm [Xu et al., 2020]. As depicted in Figure 1, the objective extends beyond the mere prediction of forthcoming driving actions (e.g., “Stop”) and includes generating a set of plausible explanations (e.g., “Red Traffic Light”). These justifications are vital for enhancing the explainability of the AD system, thereby bolstering public trust. To this end, various methods have been developed [Zablocki et al., 2022]. For example, OIA [Xu et al., 2020] utilizes an object detector to identify action-inducing objects while F-Transformer [Dong et al., 2023] employs a transformer-based [Vaswani et al., 2017] module to obtain a global scenario understanding.

Limitations. Despite considerable developments, current EAD methods suffer from two fundamental deficiencies. Firstly, there is an inadequate exploitation of the semantic information inherent in actions and explanations (referred to as categories in this work). This semantic richness can guide
the learning of more discriminative representations. For instance, the explanation “Solid Line on the Left” should direct the model to focus on the left-side lane marking, a feature frequently overlooked by object detectors and transformers in existing EAD models. Secondly, current approaches neglect the dynamic correlations among categories. These inter-category relations are imperative for avoiding inconsistencies between predicted categories and identifying categories that image feature extractors may ignore. For example, detecting a “Red Traffic Light” should trigger the “Stop” action and inherently inhibit the “Go Forward” action, potentially coupled with anticipating the “Obstacle: Person” explanation.

**Contributions.** To address these limitations, we introduce Semantic-Guided Dynamic Correlation Learning (SGDCL). SGDCL is designed to leverage semantics within categories to learn category-specific representations and model their interactions for enhanced performance. Specifically, SGDCL utilizes a semantic-guided learning module to refine features for each category. This allows each category to focus on its semantically relevant scene regions, resulting in more distinctive representations. Building on this, we use graphs derived from co-occurrence statistics of categories to link these representations. We then employ a graph neural network to explore their intricate interplay. In particular, SGDCL implements a graph attention network [Veličković et al., 2018] to dynamically assess category relevance for each sample while considering the heterogeneity of graphs. Moreover, we devise a readout function to obtain a compact graph-level embedding by combining node-level representations, facilitating a holistic understanding of the scene. To regularize model training, we introduce a novel loss function term that harnesses co-occurrence statistics of categories in a fine-grained manner.

In summary, our contributions are as follows:

- We exploit a semantic-guided learning module that directs categories to their relevant scene regions, generating more discriminative category-specific features. These tailored features accurately signal a category’s presence, enhancing prediction performance.
- We develop a dynamic correlation learning module with the pioneering usage of co-occurrence statistics for regularization. This module dynamically determines sample-specific category interactions, providing extra insights for more consistent and comprehensive predictions.
- We extensively evaluate SGDCL on two widely used benchmarks, showcasing its superiority over seven state-of-the-art baselines and a large vision-language model. Notably, SGDCL improves the performance up to 15.3% and provides interpretable attention scores, advancing the explainability of AD systems by a large margin.

2 Related Work

**Explainable Autonomous Driving.** Explainability in autonomous driving systems is pivotal to bolster human trust in self-driving vehicles [Atakishiyev et al., 2021; Omeiza et al., 2021; Goldman and Bustin, 2022; Madhav and Tyagi, 2022; Zablocki et al., 2022]. Explainable autonomous driving goes beyond driving action predictions and strives to elucidate the explanations behind the predicted actions, which has witnessed significant innovation [Cultrera et al., 2020; Koh et al., 2020; Xu et al., 2020; Jing et al., 2022; Zhang et al., 2022; Dong et al., 2023; Feng et al., 2023]. Xu et al. introduce a dataset for benchmarking prediction of actions and explanations, alongside a model based on Faster R-CNN [Ren et al., 2015] to recognize action-inducing objects [Xu et al., 2019]. To attain a comprehensive scene understanding, NLEDM [Feng et al., 2023] and F-Transformer [Dong et al., 2023] adopt a scene segmentation module and a transformer-based [Vaswani et al., 2017] architecture, respectively. Moreover, ABIM [Zhang et al., 2022] and InAction [Jing et al., 2022] consider the interrelations among traffic-related objects to improve explainability. Nonetheless, these methods overlook the semantics embedded in categories. For example, neither object detectors nor scene segmentation models focus on the lane line, a crucial feature for explanations such as “Solid Line on the Right”. To this end, our SGDCL exploits category semantics to learn category-specific representations that attend to relevant semantic regions. Furthermore, the interplay between categories, which can benefit prediction, remains unexplored. For instance, detecting a “Red Traffic Light” should inform both the “Stop” action and possible “Obstacle: Person” explanation. To model such interactions, we propose to construct graphs based on co-occurrence statistics of categories and conduct message passing on generated graphs using a graph neural network (GNN) [Wu et al., 2020].

**GNNs for Relationship Exploration.** Applying GNNs to explore relationships among multiple elements has proven effective across various fields [Chen et al., 2019a; Chen et al., 2019b; Wang et al., 2020; Ye et al., 2020; Chen et al., 2021b]. For image recognition, SSGRL [Chen et al., 2019a] employs a gated recurrent unit (GRU) for message propagation on graphs and refining node-level features. ML-GCN [Chen et al., 2019b] applies a graph convolutional network (GCN) [Kipf and Welling, 2017] to aggregate information and update node representations. However, directly applying these strategies for the joint prediction of actions and explanations yields suboptimal results due to (i) the ignored heterogeneity of nodes (i.e., action nodes and explanation nodes) and edges in the constructed category graph, (ii) the simplistic utilization of node-level features, and (iii) underexplored category correlation information. These oversights result in the neglect of essential information: (i) the differences in the interaction patterns between heterogeneous nodes, (ii) an overall understanding of the driving scenario, and (iii) the individual and collective interplay among categories. In contrast, our SGDCL (i) distinguishes edge types and corresponding interplay patterns during message passing, (ii) derives a more expressive graph-level representation from node features through a readout function, and (iii) incorporates a graph attention module to dynamically learn category correlations with the seminal implementation of fine-grained co-occurrence statistics for global regularization.

3 Method

This section elaborates on our proposed method, namely SGDCL. It begins with an explicit definition of the problem...
under study, followed by a systematic overview of our approach. Subsequently, we delve into in-depth discussions of model design and the formulation of training objectives.

### 3.1 Problem Definition

Given a frame captured by an AD vehicle’s dashboard camera (dash-cam), we aim to predict and explain the probable subsequent driving actions. Building on prior work [Xu et al., 2020], we frame this problem as a multi-task and multi-label learning task. For a given dash-cam image \( x_i \) from the input space \( \mathcal{X} \), our goal is to forecast a set of feasible actions \( \mathcal{Y} \) along with their corresponding explanations \( \mathcal{E} \). Here \( \mathcal{Y} \) and \( \mathcal{E} \) constitute the output space. Note that multiple actions for a scene (e.g., “Stop” and “Turn Left” when an obstacle is detected ahead) and multiple explanations for an action (e.g., “Obstacle: Car” and “Red Traffic Light” for the “Stop” action) are plausible. Accordingly, \( \mathcal{Y} = \{0, 1\}^{M_a} \) and \( \mathcal{E} = \{0, 1\}^{M_e} \), where \( M_a \) and \( M_e \) denote the numbers of actions and explanations, respectively. We introduce categories as an umbrella term to encompass both actions and explanations, with a total of \( M = (M_a + M_e) \) items.

### 3.2 Framework Overview

As depicted in Figure 2, SGDCL contains four key components: two encoders for the input image and categories, a semantic-guided learning (SGL) module, a dynamic correlation learning (DCL) module, and dual classifier heads. The semantic-guided learning (SGL) module, dynamic correlation learning (DCL) module, and classifier heads. The training objective of SGDCL consists of three loss terms: \( L_{act} \), \( L_{exp} \), and \( L_{cor} \).

**Encoders**

Drawing inspirations from previous work [Feng et al., 2023], we use DeepLabV3 [Chen et al., 2017] as our image encoder. The feature map for each input image \( x_i \) is \( h_i = \text{DeepLabV3}(x_i) \in \mathbb{R}^{W \times H \times D} \), where \( W, H, \) and \( D \) are the width, height, and channel number, respectively.

Since each category \( c_j \) in \( \mathcal{Y} \) and \( \mathcal{E} \) is a textual sentence, we employ the pre-trained Sentence-BERT [Reimers and Gurevych, 2019] as our language encoder. The category semantic information \( s_j \in \mathbb{R}^{d_s} \) is defined as \( s_j = \text{Sentence-BERT}(c_j), j \in \{1, \ldots, M\} \).

**Semantic-Guided Learning (SGL)**

The semantic-guided learning (SGL) module is designed to extract category-specific representations by selectively attending to semantically related image regions, as informed by category semantics. For example, left-oriented categories (e.g., action “Turn Left” and explanation “Solid Line on the Left”) should focus more on the left region of the image.

Inspired by previous work [Chen et al., 2019a], for each location \((w, h)\) in the feature map \( h_i \), we merge its image feature \( h_i^{w,h} \in \mathbb{R}^{D} \) and sentence embedding \( s_j \in \mathbb{R}^{d_s} \):

\[
h_i^{w,h} = \tanh(W_1 h_i^{w,h} \odot W_2 s_j),
\]

where \( W_1 \in \mathbb{R}^{d_s \times D} \) and \( W_2 \in \mathbb{R}^{d_s \times d_s} \) are the trainable parameter matrices, \( d_s \) is the dimension of the joint embedding, \( \tanh \) denotes the hyperbolic tangent function, and \( \odot \) represents element-wise multiplication. Subsequently, we calculate the cross-attention coefficient \( \alpha_{i,j}^{w,h} \) for the joint embedding \( h_i^{w,h} \) using a linear layer:

\[
\alpha_{i,j}^{w,h} = h_i^{w,h} W_3^{\top},
\]

where \( W_3 \in \mathbb{R}^{1 \times d_s} \) is a trainable parameter matrix. This coefficient signifies the importance of location \((w, h)\) in the \( i \)-th image for the \( j \)-th category. \( \alpha_{i,j}^{w,h} \) are then normalized over all locations using a Softmax function:

\[
\alpha_{i,j}^{w,h} = \frac{\exp(\alpha_{i,j}^{w,h})}{\sum_{w'=1}^{W} \sum_{h'=1}^{H} \exp(\alpha_{i,j}^{w',h'})},
\]

where \( \exp \) is the exponential function. Finally, we perform a weighted sum operation over all locations with normalized
cross-attention coefficients to generate the category-specific embedding \( f_{i,j} \in \mathbb{R}^D \) for category \( c_j \):

\[
f_{i,j} = \sum_{h=1}^{W} \sum_{w=1}^{H} \alpha_{i,j,h} f_{i,k,h}.
\]

This SGL module effectively directs the network’s attention to diverse regions of the image, guided by category semantics. It enables distinct category semantic contexts to yield different representations for the same input image \( x_i \). This operation is repeated for all \( M \) categories, resulting in a category-specific feature matrix \( F_i \in \mathbb{R}^{M \times D} \).

**Dynamic Correlation Learning.** The dynamic correlation learning (DCL) module exploits a graph-based approach to model intricate correlations among category representations. It leverages the inherent co-occurrence statistics of categories to construct a graph and adopts a GNN to adaptively learn high-order relationships among categories.

**Graph Construction.** A graph is represented by \( G_s = \{ V, F_s, A \} \). Here, the node set \( V \) contains \( M \) nodes, each corresponding to a category, \( F_s \) denotes the category-specific feature matrix from the SGL module. \( A \in \mathbb{R}^{M \times M} \) is the adjacency matrix, where \( A_{j,k} \) indicates the relevance between node \( j \) and node \( k \). The forthcoming discussion describes the generation of \( A \), a procedure that is foundational to DCL.

- **Co-occurrence.** We calculate the co-occurrence probabilities between categories: \( A_{j,k} = T_{jk}/T_j \), where \( T_{jk} \) is the count of co-occurrences for categories \( c_j \) and \( c_k \), and \( T_j \) is the total occurrences of category \( c_j \). \( A_{j,k} \) is the probability of encountering \( c_k \) given the presence of \( c_j \). This computation is carried out for all category pairs based on training data without additional annotation.

- **Binarization.** Directly utilizing \( A \) as the adjacency matrix may lead to suboptimal results because it imposes a uniform correlation pattern across all samples, which may not generalize well. To overcome this, our model adaptively learns the specific category relationships for each sample. The function of \( A \) is twofold: it indicates nodes’ connectivity and acts as a regularization mechanism (will be detailed in Section 3.4). Since co-occurrence statistics of categories typically follow a long-tail distribution, with infrequent co-occurrences introducing noise, we apply a binarization process to \( A \). Specifically, we set elements below the pre-defined threshold to 0 and above it to 1, resulting in \( A \). Thresholds are determined separately for action and explanation nodes, denoted by \( \gamma_1 \) and \( \gamma_2 \), respectively.

In light of the heterogeneity of the graph (nodes stemming from two different tasks), edge attributes are crucial for accurately characterizing the category interaction patterns within and across tasks. Consequently, we represent four types of directed edges using a pair of binary indicators \( r_{j,k} \):

\[
r_{j,k} = \begin{cases} 
0, & \text{if } c_j \in \mathcal{Y} \text{ and } c_k \in \mathcal{Y} \\
0, & \text{if } c_j \in \mathcal{Y} \text{ and } c_k \in \mathcal{E} \\
1, & \text{if } c_j \in \mathcal{E} \text{ and } c_k \in \mathcal{Y} \\
1, & \text{if } c_j \in \mathcal{E} \text{ and } c_k \in \mathcal{E} 
\end{cases}
\]

**Graph Neural Network.** Within our framework, we harness a Graph Attention Network (GAT) [Veličković et al., 2018] to dynamically refine sample-specific category correlations and update node representations.

For node \( j \) with input image \( x_i \), the attention coefficients \( \beta_{j,k} \) relative to the node \( k \) from its first-order neighbor set \( N_j \) (inclusive of the node itself) is determined by:

\[
\beta_{i,j,k} = \frac{\exp(\sigma(\langle f_i \rangle_{M}T_{j,k}\|g_r\rangle_{k}))}{\sum_{k \in N_j} \exp(\sigma(\langle f_i \rangle_{M}T_{j,k}\|g_r\rangle_{k}))},
\]

where \( \mathbf{W}_4 \in \mathbb{R}^{d_4 \times D} \), \( \mathbf{W}_5 \in \mathbb{R}^{d_5 \times 2} \), and \( \mathbf{W}_6 \in \mathbb{R}^{1 \times d_3} \) are the trainable parameter matrices, \( d_3 \) is the dimension of the updated node embedding, \( \sigma \) introduces LeakyReLU non-linearity, and \( g_r \) denotes the concatenation operation. We update node \( j \)'s representation by aggregating neighbor features weighted by the attention coefficients:

\[
f_{i,j} = \mathrm{ELU}(\sum_{k \in N_j} \beta_{i,j,k} f_{4,i,k}),
\]

where ELU represents the Exponential Linear Unit, and \( f_{i,j} \) is the updated representation after one round of message passing. By conducting this message passing process \( L \) times, each node is allowed to integrate information from its \( L \)-hop neighborhood, resulting in enriched category representations \( \{f_{L,j}\}_{j=1}^M \) that capture the nuanced interplay among the category semantics. Notably, the final graph attention coefficient \( \beta_{i,j,k} \) reflects the dynamic inter-category relationships between category \( c_j \) and \( c_k \) for input \( x_i \).

**Readout.** To gain a holistic understanding of the input image \( x_i \), we use a readout function to generate a graph-level embedding \( g_i \). This function concatenates all individual node representations and forwards them to a linear layer:

\[
g_i = \mathbf{W}_7(||\{f_{L,j}\}_{j=1}^M||),
\]

where \( \mathbf{W}_7 \in \mathbb{R}^{d_5 \times M d_4} \) is the trainable parameter matrix with \( d_4 \) and \( d_5 \) denoting the dimensions of the node embedding and the graph-level representation, respectively.

**Classifier Heads.**

The final stage of our model deploys two linear layers with Sigmoid activation to predict actions \( \hat{y}_i \) and explanations \( \hat{e}_i \):

\[
\hat{y}_i = \mathrm{Sigmoid}(\mathbf{W}_8 g_i),
\hat{e}_i = \mathrm{Sigmoid}(\mathbf{W}_9 g_i),
\]

where \( \mathbf{W}_8 \in \mathbb{R}^{M_{act} \times d_5} \) and \( \mathbf{W}_9 \in \mathbb{R}^{M_{exp} \times d_5} \) are the trainable parameter matrices.

### 3.4 Training Objective

To optimize our network, we adopt a multi-task loss function:

\[
\mathcal{L} = L_{act} + \lambda L_{exp} + \eta L_{cor},
\]

where hyperparameters \( \lambda \) and \( \eta \) control the impact of corresponding loss component. \( L_{act} \) and \( L_{exp} \) are the binary cross entropy losses for action and explanation prediction, respectively. Moreover, We integrate a correlation-based regularizer...
\[ \mathcal{L}_{cor} \text{ informed by the co-occurrence matrix } \bar{A} \text{ (before binarization) and the final graph attention coefficient } \beta_{i,j,k}: \]
\[
\mathcal{L}_{cor} = \frac{1}{M^2 N} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{M} (\beta_{i,j,k} - \hat{A}_{i,j,k})^2.
\]
where \( N \) denotes the number of training samples.

**Discussion.** Our approach effectively utilizes the information about category co-occurrence by distinguishing between (i) the coarse-grained binary matrix \( A \), which we input into the GAT to depict node connectivity, and (ii) the fine-grained matrix \( \bar{A} \), which we leverage within the correlation-based loss \( \mathcal{L}_{cor} \). Unlike previous methods that directly utilize the fine-grained co-occurrence probabilities \( A \) as the adjacency matrix, our approach allows for sample-specific node interactions by dynamically determining the graph attention coefficients across input images. This adaptability is important for generalization as the correlations between categories vary depending on the context. For instance, the explanation for “Stop” in one image might be “Obstacle: Car,” while for another image, it could be “Red Traffic Light.” To prevent the potential loss of informational granularity, we incorporate fine-grained information as a regularizer in our loss function.

Note that \( \mathcal{L}_{cor} \) encourages the average learned graph attention coefficients of all samples to align with the co-occurrence probabilities. This dual strategy of simultaneously capturing global and individual patterns is critical in performance improvement, as demonstrated in the following section.

### 4 Experiments

This section extensively evaluates SGDCL against seven state-of-the-art baselines and a large vision-language model on two popular benchmarks. Besides quantitative evaluation, we provide qualitative results to elucidate the reasons for SGDCL’s effectiveness. Then, a detailed ablation study is conducted to assess the contribution of individual components within SGDCL. Lastly, we evaluate the impact of crucial hyperparameters on SGDCL’s performance.

#### 4.1 Experimental Setups

**Datasets.** We conduct experiments on two commonly used datasets: BDD-OIA [Xu et al., 2020] and PSI [Chen et al., 2021a]. BDD-OIA, derived from BDD100K [Yu et al., 2020], contains 22,924 video frames, each annotated with 4 action decisions and 21 human-defined explanations. PSI includes 11,902 keyframes, each annotated with 3 actions and 29 explanations. We divide both datasets into 70% training, 10% validation, and 20% testing samples.

**Evaluation Metrics.** Since both action and explanation prediction tasks are multi-label classification problems for BDD-OIA, we use two variants of the standard F1 score metric, i.e., overall F1 (oF1) and marco-F1 (mF1), for quantitative evaluation. The oF1 averages the F1 score across the testing set:

\[
\text{Act\_oF1} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \text{F1}(y_i, \hat{y}_i),
\]

\[
\text{Exp\_oF1} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \text{F1}(e_i, \hat{e}_i),
\]

where \( Q \) is the number of testing samples. \( \text{Act\_oF1} \) and \( \text{Exp\_oF1} \) represent F1 scores for action and explanation prediction, respectively. Given the imbalanced nature of the BDD-OIA dataset, we also calculate the mF1:

\[
\text{Act\_mF1} = \frac{1}{M_{\text{act}}} \sum_{j=1}^{M_{\text{act}}} \text{F1}_j,
\]

\[
\text{Exp\_mF1} = \frac{1}{M_{\text{exp}}} \sum_{j=1}^{M_{\text{exp}}} \text{F1}_j,
\]

where \( \text{F1}_j \) is the F1 score for the \( j \)-th category.

For the PSI dataset, evaluation prediction remains a multi-label classification problem. Thus, we use \( \text{Exp\_oF1} \) and \( \text{Exp\_mF1} \) as evaluation metrics. Since action prediction is a single-label classification task, we use the overall accuracy \( \text{Act\_oAcc} \) and class-wise average accuracy \( \text{Act\_mAcc} \) as evaluation metrics. These two metrics are defined by substituting the F1 score in Eq. (2) and Eq. (3) with accuracy.

**Baselines.** To validate the effectiveness of our method, we compare it against the following competitive baselines:

- ResNet [He et al., 2016], which is pre-trained and then fine-tuned on both datasets.
- CBM [Koh et al., 2020], which exploits the concept bottleneck model to predict actions and explanations.
- OIA [Xu et al., 2020], which leverages Faster R-CNN [Ren et al., 2015] and a global context module to determine action-inducing objects.
- NLE-DM [Feng et al., 2023], which makes predictions based on the scene segmentation module.
- ABIM [Zhang et al., 2022], which captures the inter-relationship among traffic-related objects using a dual-module algorithm to predict actions and explanations.
- InAction [Jing et al., 2022], which models both explicit human annotation and implicit visual semantics for improved prediction performance.
- F-Transformer [Dong et al., 2023], which adopts a fully transformer-based structure to perform global attention.
- GPT-4V [OpenAI, 2023], which is one of the latest visual-language models. For a fair comparison, we input images with optional categories.

**Implementation Details.** We pre-train the image encoder (DeepLabV3) using a part of the BDD100K dataset [Yu et al., 2020] and then fine-tune it on both datasets. We utilize a stochastic gradient descent (SGD) optimizer with an initial learning rate of 0.001, a momentum of 0.9, a weight decay of \( 1 \times 10^{-4} \), and a batch size of 2. The hyperparameters \( \lambda \) and \( \eta \) in Eq. (1) are set to 1.5 and 0.2, respectively. For the image encoder, the output dimensions are set as \( W = 32, H = 18, \) and \( D = 25 \). The output dimension \( d_1 \) is 768 for the language encoder. In the SGL module, the joint embedding dimension \( d_2 \) is 8. For graph construction, we set \( \gamma_1 = 0.45 \) and \( \gamma_2 = 0.07 \). In GAT, the number of message passing iterations \( L \), the hidden state dimension \( d_3 \), the output node dimension \( d_4 \), and graph-level representation dimension \( d_5 \).
are set to 2, 8, 16, and 64, respectively. Additionally, we employ multi-head attention to stabilize the learning process of graph attention with 8 attention heads. For reproducibility, the source code is publicly available\(^1\).

### 4.2 Experiment Results

**Quantitative Results.** From Table 1, which presents the action and explanation prediction performance, we have the following observations: (i) SGDCL consistently outperforms all baselines by a significant margin. Compared with the best-performing baselines, SGDCL achieves performance improvements of 2.5% to 9.3% and 3.1% to 15.3% on the BDD-OIA and PSI, respectively. This proves SGDCL’s efficacy over existing state-of-the-art models since it learns category-specific representations by a cross-attention mechanism and models their dynamic interactions via a graph attention module with appropriate regularization. (ii) CBM performs better than ResNet by learning high-level concepts alongside image features. NLE-DM surpasses OIA, highlighting the benefits of using a scene segmentation module to encode images. InAction and ABIM achieve better explanation prediction results by modeling object relationships. F-Transformer, encoding global scene information, often performs best among baselines. While GPT-4V demonstrates impressive scene understanding and reasoning capabilities [Wen et al., 2023], it manifests limitations in recalling all possible actions and explanations. Even when “Turning Left” and “Turning Right” are viable actions, GPT-4V tends to be overly cautious by favoring “Stop” [Cui et al., 2023].

**Qualitative Results.** To showcase the superiority of SGDCL, we present qualitative results in Figure 3. In the first example, OIA overlooks the expected explanations: “Solid Line on the Left” and “Solid Line on the Right.” This omission leads to incorrect action predictions: “Turn Left” and “Turn Right,” which are hazardous and violate traffic rules. This mistake stems from OIA’s object detector failing to recognize the lane line. In comparison, SGDCL identifies the lane line. In comparison, SGDCL identifies the lane line. In comparison, SGDCL identifies the lane line. In comparison, SGDCL identifies the lane line. In comparison, SGDCL identifies the lane line. In comparison, SGDCL identifies the lane line. In comparison, SGDCL identifies the lane line. In comparison, SGDCL identifies the lane line. In comparison, SGDCL identifies the lane line. In comparison, SGDCL identifies the lane line. In comparison, SGDCL identifies the lane line. In comparison, SGDCL identifies the lane line. In comparison, SGDCL identifies the lane line. In comparison, SGDCL identifies the lane line.

To demonstrate how our cross-attention mechanism selectively concentrates on semantically relevant areas, we visualize the cross-attention coefficient \(\alpha_{i,j}^{wh}\) in Figure 4. We differentiate the attention driven by categories associated with left and right, highlighting areas with high values. The visualizations indicate that coefficients guided by left-related categories focus on critical information in the left region. Conversely, coefficients informed by right-related categories attend to vital information on the right.

**Ablation Study.** To ascertain the contribution of each component in SGDCL, we conduct a comprehensive ablation study. The critical components under scrutiny are semantic-guided learning (SGL), dynamic correlation learning (DCL), and correlation-based loss (CL). We also consider different network architectures and readout functions in DCL. The variants of SGDCL are as follows:

- **SGDCL w/o SGL** directly concatenates representations from the image encoder \((h_i)\) and language encoder \((s_j)\) and forwards these to the DCL module.
- **SGDCL w/o DCL** directly makes predictions based on the category-specific embedding \(f_{i,j}\).
- **SGDCL w/o CL** discards the last term in Eq. (1).
- **SGDCL w/o EA** drops the binary edge attribution \(r_{jk}\).
- **SGDCL-GCN** employs GCN [Kipf and Welling, 2017] as the graph neural network with \(\mathbf{A}\) as a constant adjacency matrix. Correspondingly, the last term in Eq (1) is always 0 and omitted.
- **SGDCL w/o Readout** directly makes predictions based on node-level embedding \(f^{li}_{i,j}\).
- **SGDCL-Mean** and **SGDCL-Max** generate graph-level embedding \(g_i\) by applying mean and max pooling on node-level representations, respectively.

The results on the BDD-OIA dataset are shown in Table 2. We observe that both SGL and DCL modules significantly enhance model performance, confirming our motivation for learning category-specific representations and modeling their relationships. Performance degradation without CL verifies its importance in training by leveraging fine-grained co-occurrence information as a global regularizer. Modeling category correlation without edge attributions leads to decreased results, indicating that considering the diversity of node and

<table>
<thead>
<tr>
<th>Dataset</th>
<th>BDD-OIA [Xu et al., 2020]</th>
<th>PSI [Chen et al., 2021a]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Act_of1</td>
<td>Act_mnF1</td>
</tr>
<tr>
<td>ResNet</td>
<td>0.601</td>
<td>0.392</td>
</tr>
<tr>
<td>CBM</td>
<td>0.661</td>
<td>0.610</td>
</tr>
<tr>
<td>OIA</td>
<td>0.734</td>
<td>0.718</td>
</tr>
<tr>
<td>NLE-DM</td>
<td>0.733</td>
<td>0.723</td>
</tr>
<tr>
<td>ABIM</td>
<td>0.722</td>
<td>0.701</td>
</tr>
<tr>
<td>InAction</td>
<td>0.714</td>
<td>0.694</td>
</tr>
<tr>
<td>F-Transformer</td>
<td>0.735</td>
<td>0.703</td>
</tr>
<tr>
<td>GPT-4V</td>
<td>0.537</td>
<td>0.436</td>
</tr>
<tr>
<td>SGDCL</td>
<td><strong>0.753</strong></td>
<td><strong>0.733</strong></td>
</tr>
</tbody>
</table>

Table 1: The comparison of action and explanation prediction performance. The best results are **bold**, and the runner-up ones are underlined.

\(^1\)https://github.com/ChengtaiCao/SGDCL
Figure 3: Qualitative comparison of action and explanation predictions between OIA and SGDCL. Detected objects in OIA are delineated with bounding boxes. The regions of significant attention and the informative generated sub-graph in SGDCL are presented.

Figure 4: Visualizations of semantic-guided learning. The regions with high cross-attention coefficient $\alpha_{i,j}$ are highlighted.

Figure 5: Hyperparameter study of SGDCL.

<table>
<thead>
<tr>
<th>Variant</th>
<th>Act_oF1</th>
<th>Exp_oF1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGDCL w/o SGL</td>
<td>0.723</td>
<td>0.528</td>
</tr>
<tr>
<td>SGDCL w/o DCL</td>
<td>0.727</td>
<td>0.533</td>
</tr>
<tr>
<td>SGDCL w/o CL</td>
<td>0.734</td>
<td>0.566</td>
</tr>
<tr>
<td>SGDCL w/o EA</td>
<td>0.728</td>
<td>0.536</td>
</tr>
<tr>
<td>SGDCL-GCN</td>
<td>0.722</td>
<td>0.515</td>
</tr>
<tr>
<td>SGDCL w/o Readout</td>
<td>0.716</td>
<td>0.419</td>
</tr>
<tr>
<td>SGDCL-Mean</td>
<td>0.736</td>
<td>0.448</td>
</tr>
<tr>
<td>SGDCL-Max</td>
<td>0.745</td>
<td>0.562</td>
</tr>
<tr>
<td><strong>SGDCL</strong></td>
<td><strong>0.753</strong></td>
<td><strong>0.582</strong></td>
</tr>
</tbody>
</table>

Table 2: Ablation study of SGDCL.

Hyperparameter Sensitivity. Figure 5 presents the sensitivity analysis results for two critical hyperparameters: $\lambda$ and $\eta$ in Eq. (1). For $\lambda$, which balances two main tasks, we vary it within the set \{0.8, 1.0, 1.2, 1.5, 2.0\}. For $\eta$, responsible for regulating the strength of the auxiliary task, we select values from the set \{0.1, 0.2, 0.3, 0.5\}. We observe that SGDCL’s performance is insensitive to these two hyperparameters, and this robustness is another advantage of SGDCL.

5 Conclusion

This work introduces SGDCL, a novel approach for explainable autonomous driving. SGDCL addresses critical shortcomings of existing methods via a semantic-guided learning module and a dynamic correlation learning module to learn category-specific features and model their interplay. Furthermore, we propose a novel loss item that leverages fine-grained co-occurrence statistics to regularize model training. Our comprehensive evaluation of two benchmarks demonstrates its effectiveness, surpassing seven state-of-the-art baselines and a large vision-language model. SGDCL improves prediction performance by a large margin and offers interpretable attention scores, enhancing the explainability and transparency of autonomous driving systems.

Acknowledgments

The work is supported in part by the Hong Kong Research Grant Council under GRF 11210622.
References


[Wang et al., 2020] Ya Wang, Dongliang He, Fu Li, Xiang Long, Zhichao Zhou, Jinwen Ma, and Shilei Wen. Multi-label classification with label graph superimposing. In...


