Causal-based Time Series Domain Generalization for Vehicle Intention Prediction

Anonymous Author(s)
Affiliation
Address
email

Abstract

Accurately predicting the possible behaviors of traffic participants is an essential capability for autonomous vehicles. Since autonomous vehicles need to navigate in dynamically changing environments, they are expected to make accurate predictions regardless of where they are and what driving circumstances they encountered. Therefore, generalization capability to unseen domains is crucial for prediction models when autonomous vehicles are deployed in the real world. In this paper, we aim to address the domain generalization problem for vehicle intention prediction tasks and a causal-based time series domain generalization (CTSDG) model is proposed. We construct a structural causal model for vehicle intention prediction tasks to learn an invariant representation of input driving data for domain generalization. We further integrate a recurrent latent variable model into our structural causal model to better capture temporal latent dependencies from time-series input data. The effectiveness of our approach is evaluated via real-world driving data. We demonstrate that our proposed method has consistent improvement on prediction accuracy compared to other state-of-the-art domain generalization and behavior prediction methods.

1 Introduction

Researchers have developed various machine learning algorithms to enhance the accuracy of intention and motion prediction tasks for autonomous vehicles [1, 2, 3, 4, 5, 6, 7, 8, 9]. Although these machine learning algorithms provide excellent prediction performance, a key assumption underlying the remarkable success is that the training and test data usually follow similar statistics. Otherwise, when test domains are unseen or Out-of-Distribution (OOD) [10], the resulting train-test domain shift leads to significant degradation in prediction performance. Domain generalization (DG) aims to build models that are designed for increased robustness to domain-shift without requiring access to target domain data. Many researchers have been working on developing DG algorithms [11, 12, 13, 14, 15, 16, 17, 18, 19, 20], but most of these models are designed for tasks such as object recognition, semantic segmentation, image classification, and disease detection. In this paper, our goal is to design a DG algorithm for vehicle intention prediction task which is different from typical DG tasks in three aspects: (1) the input is numerical time-series data instead of static image; (2) each input data point contains information of multiple agents (e.g. vehicles) with interactions involved, instead of a single agent (e.g. patient) with no interaction; (3) factors that cause domain shift are ambiguous for our task, in contrast to clear domain shift variables for other DG tasks (e.g. image rotation angle or style for image classification task). Therefore, directly applying state-of-the-art DG approaches may not be suitable for our problem setting and could result in undesirable performance. In this paper, a causal-based time series domain generalization (CTSDG) model for vehicle intention prediction is proposed. To the best of our knowledge, this is the first work trying to tackle time series domain generalization problem for vehicle intention prediction task. The effectiveness of our approach is

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

evaluated on real-world driving data under different scenarios. Our method also shows consistent improvement on prediction accuracy compared to other state-of-the-art methods.

2 Method

2.1 Notations

Let $\mathcal{X}$ be the input space and $\mathcal{Y}$ the label space, a *domain* is defined as a joint distribution P_{XY} on $\mathcal{X} \times \mathcal{Y}$. In the context of domain generalization, we assume to have access to K domains, $\mathcal{D} = \{D_k\}_{k=1}^{K}$, each associated with a joint distribution P_{XY}^{k}. In each domain, data-label pairs are sampled from a dataset $D_k = \{(\mathbf{x}_i^{(k)}, \mathbf{y}_i^{(k)})\}_{i=1}^{N_k}$ with $(\mathbf{x}_i^{(k)}, \mathbf{y}_i^{(k)}) \sim P_{XY}^{k}$, where N_k is the number of labeled data points in the k-th domain and $\mathbf{x}_i = (x_t^i)_{t=1}^{T} = ([\xi_t^{A,i}, \xi_t^{B,i}])_{t=1}^{T}$ denotes a multivariate time series with $x_t^i \in \mathbb{R}^{H \times 2}$. For a given vehicle, we use ξ to represent historical trajectories. In this work, given historical trajectories of interacting vehicle pairs (i.e. car A and car B), the domain generalization task is to learn an intention predictor f (i.e. predict interaction outcomes such as pass/yield) that generalizes well to unseen target domains $k' \notin D_k$ where $P_{XY}^{(k')} \neq P_{XY}^{(k)}$, $\forall k \in \{1, \ldots, K\}$.

2.2 Learning Invariant Representation from Causal Model

In this work, we propose a causal view of data generating process under vehicle interaction settings. More specifically, we utilize a structural causal model (SCM) based on domain knowledge to describe how interactive trajectories are generated by any vehicle pair (see Figure 1). The constructed causal structure provides us with causal relations between input features and the outcome, which is used to learn the intention predictor. The detailed meaning of each node variable is as follow: **Domain** (D) contains different combinations of map properties such as road topology, speed limit, and traffic rules; **Event** (E) denotes an observable variable relate to the event of two-vehicle interaction, which includes information such as initial interaction states and the length of interaction; **Driver** (O) denotes an unobservable variable relate to each driver's driving preferences or style, such as the aggressiveness level and whether he/she obeys traffic rules; **Causal features** (X_C) represent high-level time-related causal features derived from drivers' personal behavior O and their interaction status E, which are used by humans to label the intention; **Non-causal features** (X_{NC}) represent domain dependent features which contain not only domain-specific information D, but also event E; **Input Data** (X) is vehicle interactive trajectories that contain sequential multivariate data, and is constructed from a mixture of causal X_C and non-causal X_{NC} features; **Latent variable** (Z) denotes time-related latent representations extracted from time-series vehicle interaction data; **Label** (Y) is the vehicle intention label.

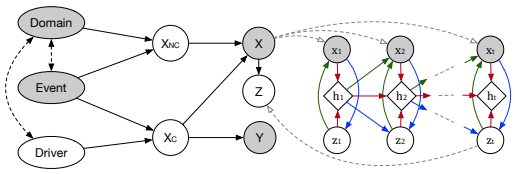


Figure 1: Illustration of the overall CTSDG framework. Observed quantities are shown as shaded nodes; nodes of latent quantities are transparent. A directed black edge (→) denotes a causal relationship; dashed bidirectional edges (◄··►) denote correlation; hollow dashed arrows (··▷) denote an inclusion relationship. Blue lines (→) denote the inference process, $q(z_t|x_{\leq t}, z_{<t})$; green lines (→) represent the generation process, $p(x_t|z_{\leq t}, x_{<t})$; red lines (→) show the recurrence process where h_t is informed by h_{t-1}, which is informed by z_{t-1} and x_{t-1}.

Invariance Condition. According to Figure 1, X_C is the node that causes Y, and by d-separation [21], the intention label is independent of domain conditioned on X_C, $Y \perp\!\!\!\perp D|X_C$[1]. In other word, if such a X_C can be found, then the distribution of Y conditional on X_C is invariant under transferring from the source domains to the target domains. Therefore, our intention prediction task is to learn y as $h(\mathbf{x}_c)$ where $h : \mathcal{C} \to \mathcal{Y}$. However, since X_C is unobserved, we need to learn y through observed trajectories X. Specifically, we utilize a representation function $q : \mathcal{X} \to \mathcal{Z}$ to map the input space to a latent space, and a hypothesis function $\phi : \mathcal{Z} \to \mathcal{C}$ to map the latent space to X_C. Together,

[1]The notation $Y \perp\!\!\!\perp D|X_C$ stands for the conditional independence relationship $P(Y = y, D = k|X_C = \mathbf{x}_c) = P(Y = y|X_C = \mathbf{x}_c)P(D = k|X_C = \mathbf{x}_c)$.

$h(\phi(q(\mathbf{x})))$ leads to the desired intention predictor $f : \mathcal{X} \rightarrow \mathcal{Y}$ and the corresponding prediction loss can be written as:

$$\mathcal{L}_y = \mathcal{L}_\alpha(h(\phi(q(X))), Y), \tag{1}$$

where $\mathcal{L}_\alpha$ is the classification loss such as a binary or categorical cross-entropy. In addition, by d-separation, X_C also needs to satisfy an invariance condition: $X_C \perp\!\!\!\perp D|\{E, O\}$, which means X_C does not change with different domains when both event and driver information remain the same. However, driver information is unobservable and in many dataset there may not be an exact match based on a same interaction event across domains. Alternatively, we assume that the distance over X_C between same-class inputs from different domains is bounded, which provides an alternative invariance condition that is consistent with the conditional independencies of X_C. Therefore, we would like to minimize the following objective along with the prediction loss:

$$\mathcal{L}_r = \sum_{\Omega(\mathbf{x}_j, \mathbf{x}_m)=1; k_j \neq k_m} \mathcal{L}_\beta(\phi(q(\mathbf{x}_j^{(k_j)})), \phi(q(\mathbf{x}_m^{(k_m)}))), \tag{2}$$

where $\mathcal{L}_\beta$ is the distance metric such as ℓ_2, and $\Omega : \mathcal{X} \times \mathcal{X} \rightarrow \{0, 1\}$ is a match function such that pairs having $\Omega(\mathbf{x}_j, \mathbf{x}_m) = 1$ have low difference in their causal features.

Contrastive Representation Learning. To optimize $\mathcal{L}_r$, we first need to learn a proper match function Ω used in Eq. (2). Specifically, we optimize a contrastive representation learning loss that minimizes distance between same-class inputs from different domains. We regard positive matches as two inputs from the same class but different domains, and negative matches as pairs with different classes. Then the loss function for every positive match pair (j, m) in a sampled mini-batch $\mathcal{B}$ is defined as:

$$\ell_{con}^{j,m} = -\log \frac{\exp(s_{cos}(\mathbf{x}_j, \mathbf{x}_m)/\tau)}{\exp(s_{cos}(\mathbf{x}_j, \mathbf{x}_m)/\tau) + \sum_{i=1, y_i \neq y_j}^{|\mathcal{B}|} \exp(s_{cos}(\mathbf{x}_j, \mathbf{x}_i)/\tau)}, \tag{3}$$

where $s_{cos}(\mathbf{x}_a, \mathbf{x}_b) = \phi(q(\mathbf{x}_a))^T \phi(q(\mathbf{x}_b))/\|\phi(q(\mathbf{x}_a))\|\|\phi(q(\mathbf{x}_b))\|$ is the inner product of two ℓ_2-normalized vectors, τ is a temperature scaling parameter, and $|\mathcal{B}|$ is the batch size. Inspired from [18], for this unsupervised contrastive learning process, we initialize Ω with a random match based on classes and keep updating Ω by minimizing the contrastive loss (3) until convergence.

2.3 Capturing Temporal Latent Dependencies

From our proposed causal model in Fig.1, we also observe that $Z \perp\!\!\!\perp D|X$, meaning the latent representation is independent of domain D conditioned on input data X. Therefore, by learning the representation function q, we are able to extract a domain-invariant latent variable that represents the input space. Since the input is time-series data, the learned latent variable is expect to capture temporal latent information in the data. To explicitly model the dependencies between latent random variable across time steps, we integrate Variational Recurrent Neural Networks (VRNN) [22] into our CTSDG model. The VRNN contains a VAE at every time step and these VAEs are conditioned on previous auto-encoders via the hidden state variable h_{t-1} of an RNN. In general, the objective function become a timestep-wise variational lower bound:

$$\mathcal{L}_v = \mathbb{E}_{q(z^i_{\leq T}|x^i_{\leq T})}[\sum_{t=1}^{T}(-\mathrm{KL}(q(z^i_t|x^i_{\leq t}, z^i_{\leq t})||p(z^i_t|x^i_{<t}, z^i_{<t})) + \log p(x^i_t|z^i_{\leq t}, x^i_{\leq t}))], \tag{4}$$

where $q(z^i_t|x^i_{\leq t}, z^i_{\leq t})$ is the inference model, $p(z^i_t|x^i_{<t}, z^i_{<t}))$ is the prior, $p(x^i_t|z^i_{\leq t}, x^i_{\leq t})$ is the generative model, and $\mathrm{KL}(\cdot||\cdot)$ refers to Kullback-Leibler divergence.

2.4 Implementation

Combining the joint optimization problem of equation (1), (2), and (4) leads to our CTSDG model. The complete objective function we aim to optimize is:

$$\mathcal{L} = \mathcal{L}_y + \gamma \mathcal{L}_v + \lambda \mathcal{L}_r, \tag{5}$$

where γ and λ are hyperparameters that control training balance among three losses for better performance. In order to prevent the classification loss from interfering learning domain-invariant representations, we learn the match function Ω before optimizing the overall objective function (5). Also, since the causal feature X_C is a function of latent variable Z which is extracted by the recurrent latent variable model, the objective function (4) needs to be jointly optimized while learning the matching function using Eq.(3).

Table 1: Domain generalization results with prediction accuracy (%).

	Traditional Baseline Methods							Behavior Prediction Methods			Ours
Source / Target	ERM [23]	IRM [24]	CCSA [25]	Mixup [20]	DANN [26]	C-DANN [16]	VRADA [27]	DESIRE [7]	SGAN [28]	NRI [29]	CTSDG
FT-1,2 / FT-3	75.97 (2.01)	82.68 (3.91)	78.03 (2.26)	60.70 (16.61)	77.47 (4.69)	81.00 (4.87)	76.35 (10.34)	75.23 (3.27)	74.46 (2.36)	76.34 (1.67)	86.03 (1.48)
FT-1,3 / FT-2	97.28 (0.82)	93.28 (3.83)	96.68 (1.42)	94.05 (3.15)	95.66 (0.77)	95.06 (3.48)	96.43 (0.51)	95.32 (0.38)	94.98 (0.96)	97.01 (0.53)	96.85 (0.15)
FT-2,3 / FT-1	98.38 (0.72)	98.76 (0.20)	98.29 (0.62)	91.56 (7.81)	99.23 (0.34)	99.19 (0.45)	98.12 (0.97)	96.85 (0.63)	99.10 (0.13)	98.77 (0.80)	98.85 (0.13)
Average	90.54	91.58	91.00	82.10	90.79	91.75	90.30	89.13	89.51	90.71	**93.91**
FT-1,2 / ZS	72.71 (3.02)	75.60 (3.99)	74.64 (2.61)	69.08 (6.25)	77.05 (4.83)	76.40 (5.55)	83.81 (3.57)	65.32 (2.34)	78.02 (5.34)	76.31 (4.49)	84.06 (3.54)
FT-1,3 / ZS	80.67 (2.74)	80.43 (1.92)	80.92 (5.63)	79.23 (1.50)	82.37 (2.54)	84.06 (4.35)	77.54 (6.91)	70.77 (4.24)	78.26 (4.75)	77.23 (4.67)	84.78 (3.49)
FT-2,3 / ZS	57.97 (0.73)	62.80 (1.67)	61.35 (2.74)	60.62 (11.55)	71.50 (13.87)	69.56 (10.67)	68.11 (9.25)	74.66 (6.03)	75.43 (1.51)	75.01 (2.91)	77.54 (3.53)
FT-1,2,3 / ZS	80.43 (3.32)	76.33 (3.72)	78.02 (3.64)	70.53 (15.92)	82.61 (3.16)	82.37 (2.54)	85.02 (2.93)	73.43 (4.93)	83.57 (2.93)	84.10 (3.22)	85.51 (2.66)
Average	72.95	73.79	73.73	69.87	78.38	78.10	78.62	71.04	78.82	78.16	**82.97**

3 Experiments

The experiment is conducted on the INTERACTION dataset [30], where two different scenarios are utilized: USA_Roundabout_FT (an eight-way roundabout in the US) and CHN_Merging_ZS (a highway merging scenario in China). In this work, a domain is defined as a driving region that contains a unique combination of road topology, traffic rules, and speed limit information. Three domains are extracted from the FT scenario and one domain is extracted from the ZS scenario. We compare our method with ten state-of-the-art methods, including seven traditional baseline methods for domain generalization tasks and three methods for behavior prediction tasks. To evaluate the domain generalization performance, we follow the typically used *leave-one-domain-out* protocol [31].

Evaluation on USA_Roundabout_FT. We first train all models on different combinations of source domains extracted in FT and report the test results on the remaining target domain in FT. From Table 1, we observe that our method achieves an average prediction accuracy of 93.91% under all three train-test domain pairs and increases the accuracy by 2.54% compared to the best-performed baseline model. ERM and NRI method slightly outperform CTSDG when FT-2 is the target domain, but their accuracy is 10.24% and 11.26% less than our method when tested on FT-3, respectively. As seen, our method has consistent prediction accuracy regardless of testing domains. The standard deviations of CTSDG are also lower than those of all other baseline methods, which further shows our model's efficacy as it converges to more stable local optima.

Evaluation on CHN_Merging_ZS. When we compare the domain in ZS with domains in FT, we notice larger difference in road topology as well as in speed limit. In other word, the joint distribution shift is more severe between domains from two different scenarios (i.e. ZS and FT) than domains within scenario FT, which makes domain generalization more challenging. Therefore, to evaluate the generalizability of each model under large domain shift, we select different domain combinations in FT for training and the single domain in ZS for testing. From Table 1, we observe that CTSDG consistently outperforms all baseline models across all source-target pairs with accuracy $1\% \sim 6\%$ higher. The average prediction accuracy of our method increases by 5% compared to the best-performed baseline model. Also, when the model is trained using all three domains in FT, where the source domains have more diversity, learning based methods that do not specifically consider domain generalization approaches (i.e. ERM, DESIRE, SGAN, and NRI) have reasonable performance. However, the performance of ERM and three behavior prediction methods degrade drastically when training data are limited and lack of diversity. According to the results, we show the diminishing performance of traditional machine learning model to out-of-training distribution regimes and our proposed method is able to mitigate such problem while consistently outperforms other state-of-the-art methods.

4 Conclusions

In this paper, we present a causal-based time-series domain generalization model (CTSDG) to address domain generalization problem for vehicle intention prediction tasks. We show that CTSDG has consistent improvements in terms of prediction accuracy compared to other state-of-the-art domain generalization and behavior prediction methods using real-world driving data. In the future, we expect to extend our work to address trajectory prediction problems and consider generalization of varying number of interacting vehicles.

References

[1] Hongyi Zhang, Andreas Geiger, and Raquel Urtasun. Understanding high-level semantics by modeling traffic patterns. In *Proceedings of the IEEE international conference on computer vision*, pages 3056–3063, 2013.

[2] Chiyu Dong, John M Dolan, and Bakhtiar Litkouhi. Intention estimation for ramp merging control in autonomous driving. pages 1584–1589, 2017.

[3] Jens Schulz, Constantin Hubmann, Julian Löchner, and Darius Burschka. Multiple model unscented kalman filtering in dynamic bayesian networks for intention estimation and trajectory prediction. pages 1467–1474. IEEE, 2018.

[4] Alex Zyner, Stewart Worrall, and Eduardo Nebot. Naturalistic driver intention and path prediction using recurrent neural networks. 2019.

[5] Sergio Casas, Wenjie Luo, and Raquel Urtasun. Intentnet: Learning to predict intention from raw sensor data. In *Conference on Robot Learning*, pages 947–956. PMLR, 2018.

[6] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Benjamin Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, et al. Tnt: Target-driven trajectory prediction. *arXiv preprint arXiv:2008.08294*, 2020.

[7] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 336–345, 2017.

[8] Nicholas Rhinehart, Rowan McAllister, Kris Kitani, and Sergey Levine. Precog: Prediction conditioned on goals in visual multi-agent settings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2821–2830, 2019.

[9] Yuexin Ma, Xinge Zhu, Sibo Zhang, Ruigang Yang, Wenping Wang, and Dinesh Manocha. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6120–6127, 2019.

[10] Angelos Filos, Panagiotis Tigkas, Rowan McAllister, Nicholas Rhinehart, Sergey Levine, and Yarin Gal. Can autonomous vehicles identify, recover from, and adapt to distribution shifts? In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2020.

[11] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pages 2551–2559, 2015.

[12] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2018.

[13] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[14] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. *Advances in Neural Information Processing Systems*, 31:998–1008, 2018.

[15] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019.

[16] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 624–639, 2018.

[17] Chang Liu, Xinwei Sun, Jindong Wang, Haoyue Tang, Tao Li, Tao Qin, Wei Chen, and Tie-Yan Liu. Learning causal semantic representation for out-of-distribution prediction. *arXiv preprint arXiv:2011.01681*, 2020.

[18] Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. *arXiv preprint arXiv:2006.07500*, 2020.

[19] Shruti Tople, Amit Sharma, and Aditya Nori. Alleviating privacy attacks via causal learning. In *International Conference on Machine Learning*, pages 9537–9547. PMLR, 2020.

[20] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[21] Judea Pearl. *Causality*. Cambridge university press, 2009.

[22] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. *arXiv preprint arXiv:1506.02216*, 2015.

[23] Vladimir Koltchinskii. *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems: Ecole d'Eté de Probabilités de Saint-Flour XXXVIII-2008*, volume 2033. Springer Science & Business Media, 2011.

[24] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[25] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5715–5725, 2017.

[26] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[27] Sanjay Purushotham, Wilka Carvalho, Tanachat Nilanon, and Yan Liu. Variational recurrent adversarial deep domain adaptation. In *ICLR (Poster)*, 2017.

[28] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018.

[29] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, pages 2688–2697. PMLR, 2018.

[30] Wei Zhan, Liting Sun, Di Wang, Haojie Shi, Aubrey Clausse, Maximilian Naumann, Julius Kummerle, Hendrik Konigshof, Christoph Stiller, Arnaud de La Fortelle, et al. Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. *arXiv preprint arXiv:1910.03088*, 2019.

[31] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

[32] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *arXiv preprint arXiv:1805.12018*, 2018.

[33] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[34] Nachiket Deo and Mohan M Trivedi. Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1179–1184. IEEE, 2018.

[35] Brian D Ziebart, Nathan Ratliff, Garratt Gallagher, Christoph Mertz, Kevin Peterson, J Andrew Bagnell, Martial Hebert, Anind K Dey, and Siddhartha Srinivasa. Planning-based prediction for pedestrians. pages 3931–3936, 2009.

[36] Hien Q Dang, Johannes Fürnkranz, Alexander Biedermann, and Maximilian Hoepfl. Time-to-lane-change prediction with deep learning. pages 1–7. IEEE, 2017.

[37] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

[38] Masashi Sugiyama and Motoaki Kawanabe. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press, 2012.

[39] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

[40] Peter W Battaglia, Razvan Pascanu, Matthew Lai, Danilo Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. *arXiv preprint arXiv:1612.00222*, 2016.

[41] Victor Bapst, Alvaro Sanchez-Gonzalez, Carl Doersch, Kimberly Stachenfeld, Pushmeet Kohli, Peter Battaglia, and Jessica Hamrick. Structured agents for physical construction. In *International Conference on Machine Learning*, pages 464–474. PMLR, 2019.

[42] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[43] Adam Santoro, David Raposo, David GT Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. *arXiv preprint arXiv:1706.01427*, 2017.

[44] Fabian Poggenhans, Jan-Hendrik Pauls, Johannes Janosovits, Stefan Orf, Maximilian Naumann, Florian Kuhnt, and Matthias Mayr. Lanelet2: A high-definition map framework for the future of automated driving. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1672–1679. IEEE, 2018.

[45] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[46] Tung Phan-Minh, Elena Corina Grigore, Freddy A Boulton, Oscar Beijbom, and Eric M Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14074–14083, 2020.

Supplemental Materials

A Related Works

Domain generalization aims to generalize models to unseen domains without knowledge about the target distribution during training. Different methods have been proposed for learning generalizable and transferable representations. A commonly-used strategy is to extract task-specific but domain-invariant features [11, 12, 16, 17, 18, 19]. Ghifary et al. [11] learn multi-task auto-encoders to extract invariant features which are robust to domain variations. Li et al. [12] extend adversarial autoencoders by imposing maximum mean discrepancy measure to align multi-domain distributions. Li et al. [16] consider the conditional distribution of label space over input space, and minimize discrepancy of a joint distribution. Recently, several works [17, 18, 19] utilize causal structures to learn invariance factors, which have shown better model generalizability on unseen data, especially on data from different distributions than the train distribution.

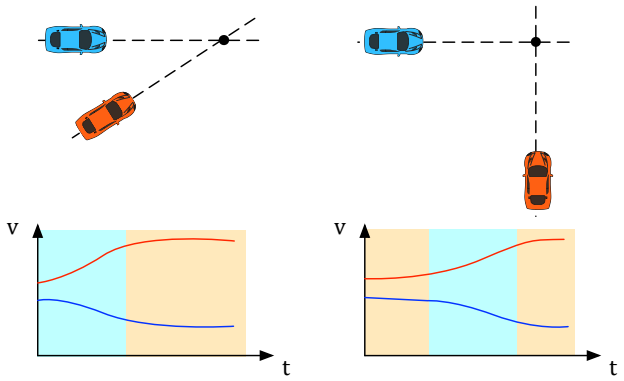
Dataset augmentation is another strategy that focuses on creating more out-of-domain samples by adversarially perturbing samples [32], leveraging self-supervised signals [15], or utilizing convex combinations of input and output data [20]. These data augmentation methods have shown effective for model generalization, but invalid samples could be generated and degrade the model performance if input data is vehicle trajectories that need to satisfy feasibility constraints. Meta-learning can also be applied in domain generalization, by creating meta-train and meta-test sets, and training a model using the meta-train set in such a way to improve the performance on the meta-test set [33, 13, 14]. While our proposed method can achieve promising results using standard training schemes, combining meta-learning with our approach could be an interesting future direction.

Intention prediction in autonomous driving aims to predict high-level behaviors of road entities, which results are often used in downstream trajectory prediction modules. For example, works such as [34], [35] and [36] demonstrate that by predicting goal states and assuming that agents navigate toward those goals by following some optimal or learned trajectories, the accuracy of prediction can be improved. Therefore, accurate intention prediction algorithm is essential for autonomous vehicles to smoothly navigate in dynamic environments. Zhang et al. [1] propose to model high level semantics in the form of traffic patterns through a generative model. Dong et al. [2] use a probabilistic graphical model to estimate the pass or yield probability under merging scenarios. Schulz et al. [3] utilize the dynamic bayesian network method for routing prediction at intersections. Zyner et al. [4] obtain a probability distribution over all possible exit branches for a vehicle driving in a roundabout using recurrent neural network. Works such as [5] and [6], use more complication ML models to prediction vehicle's intention. Due to the complexity of the real-world and its ever-changing dynamics, the deployed autonomous vehicles inevitably face novel situations and should be able to cope with them. However, it has been repeatedly demonstrated that the reliability of ML models degrades radically when they are exposed to novel settings (i.e., under a shift away from the distribution of observations seen during their training) due to their failure to generalize, leading to catastrophic outcomes [37, 38]. In [10], Filos et al. also demonstrate that several state-of-the-art behavior prediction models for autonomous driving fail to generalize under arbitrary domain shift. Therefore, in this paper, we aim to address the domain generalization problem for vehicle intention prediction task and provide an effective solution to enhance reliability of autonomous vehicles. We hope our work can inspire further research in the field of improving generalizability for prediction algorithms in autonomous driving industry.

B Methodology

B.1 Illustrate the problem context through a toy example

Consider two vehicles interact with each other under two different domains. For the ease of understanding, we simplify the problem by temporarily making the following assumptions for the two interaction cases in Figure 2: (1) The same road priorities; (2) The same two drivers interact with each other; (3) The same initial states for both vehicles. However,there are two major differences between two cases: (1) Different topological road structure of two intersected roads; (2) Two drivers have different driving styles (i.e. we let the driver in red car be more aggressive than the driver in blue car), which information is actually unavailable when making predictions.



(a) case 1 (b) case 2

Figure 2: Illustration of our design motivation through a toy example.

Based on previous assumptions and settings, for a given interaction period, a plausible simulated velocity profile of two vehicles is shown in the second row of Fig. 2 for both cases. The goal is to predict the intention of any selected vehicle, which can be the intent of one car passes or yields the other car. According to the plots, whether the red car yields or passes the blue car doesn't rely much on the domain information but depends more on two drivers' personal information (e.g. driving style) and their initial states. Specifically, the geometry of the roads could affect several factors such as when two vehicles will start noticing each other, when the road negotiation will begin, and when drivers will agree on who should go first. All these factors could then influence the velocity profiles of two vehicles but won't have much effects on the intention of two vehicles when the aforementioned assumptions hold. In other words, if two vehicles encounter each other twice under different domains, as long as the three assumptions are true, the aggressive driver will always prefer passing than yielding the less aggressive driver. The intention is determined by features that relate to two drivers' information and their internal relations (e.g. highlighted in cyan), instead of those that relate to domain itself (e.g. highlighted in orange). Hence, we construct a model for the data collection process that assumes each input data (i.e. vehicle trajectories) is constructed from a mix of causal and non-causal features. We consider domain as a main intervention that changes the non-causal features of an input data, and propose that an ideal intention predictor should depends only on the causal features.

Although the aforementioned assumptions will not hold for real-world driving data, we are still able to assume each data is constructed from both causal and non-causal features. However, it becomes unclear what are the causal and non-causal features. Therefore, in this work, we construct a structural causal model and propose to learn an invariant representation for domain generalization based on invariant conditions identified from the causal model.

B.2 Model Insights

According to [39], causality, with its focus on representing structural knowledge about the data generating process that allows interventions and changes, can contribute towards understanding and resolving some limitations of current machine learning methods. In fact, machine learning models that incorporate or learn structural knowledge of an environment have been shown to be more efficient and generalize better [40, 41, 42, 43]. For human drivers, the way they interact with each other under different domains should also processes some invariant structural relations, since they can quickly adapt their driving skills to novel scenarios. Therefore, we construct a structural causal model (SCM) for vehicle intention prediction task to learn an invariant representation of input driving data for domain generalization.

The domain generalization setting assumes the existence of domain-invariant patterns in the inputs (e.g. semantic features), which can be extracted to learn a predictor that performs well across seen and unseen domains. Moreover, causal predictive models are shown to have great generalizability since their output depends on stable relationships between input features and the outcome instead of associations between them. Therefore, in this work, we construct a structural causal model inspired

from data generation process and propose to learn an invariant representation from temporal latent dependencies of input data for domain generalization, where the learning scheme is based on an invariant condition identified from the causal model. To be more specific, we assume each input data is constructed from a mix of inherent (causal) and domain-dependent (non-causal) features. Consequently, the captured temporal latent dependencies from sequential data also contain both causal and non-causal features.

To generate a pair of interactive trajectories from two vehicles, data-generating process first samples two drivers' driving preferences (O) and a domain (D) that could be correlated to each other shown with dashed arrow in Fig. 1 (e.g. driver behavior may vary from country to country). Then, an interaction event (E) is sampled, which could correlate to domain information (e.g., the geometry of the roads may affect when two vehicles can notice/observe each other). The domain information is correspond to non-causal features (X_{NC}) and the driver information are correspond to causal features (X_C). Part of the event information, such as initial speed or location of two vehicles, are correspond to causal features which will influence the intention label. Another part of the event information, such as when the road negotiation begins, are correspond to non-causal features. Altogether, the causal and non-causal features construct the interactive trajectories (X) from where the time-related latent information (Z) are captured.

We further utilize D-separation to analyze the formulated structural causal model as it can: (1)connects causal relations and probabilistic independence; (2)computes all and only those independence and conditional independence relations that hold for all values of the parameter for any directed graph. At the lowest level of the causal hierarchy — association — we have discovered DAGs and d-separation as a powerful tool to reason about conditional (in)dependencies between variables.

B.3 Overall Algorithm

The pseudocode for our proposed CTSDG method is shown in Algorithm 1. We first initialize Ω as a random match function, where given each data point, we randomly select another data point with the same class from another domain. During each training epoch, we constantly update Ω based on the currently learned hypothesis function ϕ. Specifically, given any data point x, we find the nearest neighbour of x among other data points with the same class from another domain by calculating their distance using $\mathcal{L}_\beta$ defined in (2). After certain number of training epoch is reached, we further optimize the overall objective function (5) until convergence. Note that when calculating $\mathcal{L}_r$ in (2), we apply the updated match function Ω to select match pairs.

Algorithm 1: Causal-based time-series domain generalization (CTSDG)

Input : Source training domain $\mathcal{D} = \{D_k\}_{k=1}^K$; hyperparameters $\tau, \gamma, \lambda > 0$
Output : Intention Predictor $f : \mathcal{X} \to \mathcal{Y}$

```
Randomly split source domains D into disjoint train D_tr and validation D_val;
Initialize Ω as a random match function;
while not converged do
    Sample a mini-batch from D_tr;
    if epoch < threshold then
        Minimize (3) + γ(4);
        Update Ω;
    else
        Minimize (5);
    end
end
```

C Related Theory

C.1 D-separation

Here we provide some background concepts related to d-separation used in Section 2.3 for analyzing our causal graph.

Definition 1 ***(d-connection)****: If G is a directed graph in which X, Y and Z are disjoint subsets of nodes, then X and Y are d-connected by Z in G if and only if there exists an undirected path U between*

X and Y such that for every collider C on U, either C or a descendent of C is in Z, and no non-collider on U is in Z.

Definition 2 ***(collider)****: For any path between two nodes, a collider is a node where arrows of the path meet head-to-head.*

Definition 3 ***(d-separation)****: X and Y are d-separated by Z in G if and only if they are not d-connected by Z in G, denoted* $(X \perp\!\!\!\perp Y)|Z$.

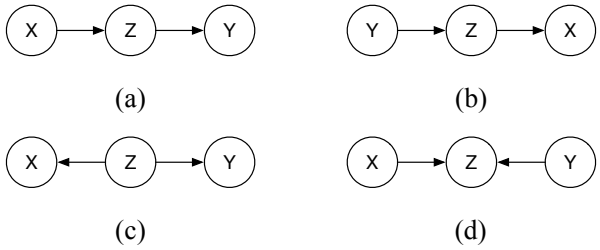


Figure 3: Four DAGs for three linked variables. The first two (i.e. (a) and (b)) are called chains; (c) is a fork; (d) is a collider. If these were the whole of the graph, we would have $X \not\perp\!\!\!\perp Y$ and $X \perp\!\!\!\perp Y|Z$. For the collider, however, we would have $X \perp\!\!\!\perp Y$ while $X \not\perp\!\!\!\perp Y|Z$.

C.2 Worse-case Generalization Error

As mentioned in Suppl.A, utilizing causal structures to learn invariance factors has shown better model generalizability on unseen data and we also demonstrate desired performances of our causal-based model in Section 3. Theoretically (i.e. Theorem 1), the worse-case generalization error for a causal model is less than or equal to that for the associational model (i.e. model that directly map from X into Y). Detailed proof can be found in Suppl.A.2 in [19].

Theorem 1. *Consider a causal model* $h^{min}_{c,D} : X_c \to Y$ *and an associational model* $h^{min}_{a,D} : X \to Y$ *trained on a dataset* $D \sim P(X, Y)$ *with loss* L*. Let* $(x, y) \in D$ *and* $(x', y') \notin S$ *be two input instances such that they share the same true labelling function on the causal features,* $y \in P(Y|X_c = x)$ *and* $y' \in P(Y|X_c = x')$*. Then, the worst-case generalization error for the causal model on such* x' *is less than or equal to that for the associational model.*

$$\max_{x \in D, x'} L_{x'}(h^{\text{min}}_{c,D}, y) - L_x(h^{\text{min}}_{c,D}, y) \leq \max_{x \in D, x'} L_{x'}(h^{\text{min}}_{a,D}, y) - L_x(h^{\text{min}}_{a,D}, y) \tag{6}$$

C.3 Justification of Minimizing Contrastive Loss

According to the invariance condition derived in Section 2.2, the idea function we want to optimize is $f^* = \arg\min_h \mathbb{E}[l(y, h(\mathbf{x}_c))]$ assuming $\mathbf{x}_c$ is known. However, as X_C is unobservable, we need to alternatively minimize the objective function in Eq.(5). Inspired from Theorem 2 in [18], we are able to derive the following theorem under our problem setting. We can observe that whether X_C will be returned by the optimization largely depends on δ_c. Specifically, if a dataset has low δ_c, then the learned representation will likely to be closer to the real X_C. Alternatively, if a dataset has high δ_c, the matching condition loses any discriminative power and thus we need to additionally learn a matching function ω to minimize δ_c. Therefore, minimizing the contractive loss (Eq.(3)) is necessary in our problem. Detailed proof or the original theorem can be found in Suppl.A.6 in [18].

Theorem 2. *Assume training domains such that for any two same-class inputs* $\mathbf{x}^{(k)}_j$ *and* $\mathbf{x}^{(k')}_j$ *from domains* k *and* k'*, dist*$(x^{(k)}_{ac,j}, x^{(k')}_{ac,m}) \leq \delta_{ac}$ *where* x_{ac} *is any non-causal feature. Further, assume that the distance over* X_C *between same-class inputs from different domains is bounded: dist*$(x^{(k)}_{c,j}, x^{(k')}_{c,m}) \leq \delta_c$ *and* $\delta_c < \delta_{ac}(\delta_c, \delta_{ac} \in \mathbb{R}^+)$*. Then for some* γ *and* λ*, a loss-minimizing classifier for the loss from Eq.(5) is the true function* f^**, given a P-admissible loss function and a finite number of domains* k *with* $K \to \infty$ *in each domain.*

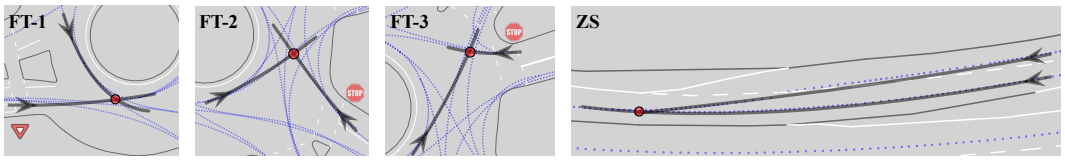


Figure 4: Illustration of selected domains for driving scenarios. Three domains selected from USA_Roundabout_FT are denoted as FT-1, FT-2, and FT-3. The one domain selected from CHN_Merging_ZS is denoted as ZS. Each black arrow line represents a reference path and red circles represent intersect points of reference paths.

D Experimental Details

D.1 Data Information

As mentioned in Section 3, the three domains (i.e. FT-1,2,3) are selected from a roundabout scenario and the domain ZS is selected from a highway scenario (see Fig.4). All data are collected by a drone from bird's-eye view with 10 Hz sampling frequency. Information such as reference paths and traffic regulations can be directly extracted from high definition maps that are in lanelet [44] format. The detailed information of each domain can be found in Table 2, where we observe that any two domains are different in at least two out of five aspects. The *intersect angle* refers to the angle between two reference paths[2]. The *average length* refers to the average length of two reference paths.

Instead of Cartesian coordinate, we utilized the Frenét Frame to represent vehicle's state. The vehicle motion in the Frenét Frame can be represented with the longitudinal position along its reference path $s(t)$, and lateral deviation to the reference path $d(t)$. Therefore the vehicle state at time step t can be defined as $\xi_t = (s(t), d(t))$. Note that the reference path of a vehicle will change according to the road it is current driving on. The origin of the reference path can be defined differently according to different objectives and each reference path will have its own Frenét Frame. Since we are dealing with interaction between two vehicles, we define the origin as the intersection point (red circles in Fig. 4) of their reference paths.

Table 2: Detailed information of each domain.

	Road Topology		Speed Limit	Traffic Rule	Location
	intersect angle	average length			
FT-1	$\approx 45°$	medium	25 mph	yield sign	USA
FT-2	$\approx 90°$	medium	25 mph	stop sign	USA
FT-3	$> 90°$	short	25 mph	stop sign	USA
ZS	$\approx 10°$	long	50 mph	zipper merging	China

We manually select the interaction data from the dataset. Since we focus on two-vehicle interaction situations, we need to make sure the two vehicles are highly interactive with each other and both vehicles' behaviors should not be influenced by any other vehicles. Also, for ease of interpretation, we balance the data such that there are equal number of interactive trajectory data per class in each domain. We select a total number of 1354 time-series data from three domains in USA_Roundabout_FT and 138 time-series data from the single domain in CHN_Merging_ZS.

In terms of the labeling process, since we have the entire interaction trajectory of two vehicles, we are able to obtain the final outcome directly based on which vehicle passes the conflict point first. We categorize the intention for two interacting vehicles as either pass or yield and consider only the period that two vehicles have high interaction. In order to make sure that selected vehicle pairs are highly interactive and their behaviors are not influenced by other surrounding vehicles, we manually examine and filter the data. The are two reasons why we filter out the data that two vehicle are far from each other. Firstly, it is difficult to obtain true labels of drivers' intentions in such situation and their intentions may not simply be pass/yield. Secondly, if two vehicle are far from each other, there will be a large chance that they either haven't started interacting with each other or may interacting

[2] A traffic-free reference path is normally obtained from road's centerline.

with other vehicles. It is worth to mention that our proposed approach can be adapted to multi-class classification with more subtle interaction patterns, but it might require human labeling.

D.2 Implementation

For domain generalization tasks, test/target domains are not accessible during training. Therefore, we randomly sample 20% of data from source domains to construct our validation set which is used for hyperparameter tuning and use the remaining source domain data for training. We use the Adam optimizer and run all models for 500 epochs with learning rate of $1e{-}3$. During training, we use a mini batch of 16 samples and apply early stopping criteria that the model does not experience a decrease in the validation loss for 20 epochs. The temperature scaling parameter τ is set to 0.05, and the two control parameters γ and λ are set to $1e-4$ and 1, respectively. For results of each model, we report the average and standard deviation over 3 independent runs.

Input dimension of each data is 10×4, where we consider the past 1s of interactive trajectories with 10Hz sampling frequency and concatenate the states of two vehicles. Output classifer dimension is 2 corresponding to the possible number of intentions. Network modules in our model comprises: an encoder network φ_τ^{enc}, which output parameters define the inference model $q(z_t^i|x_{<t}^i, z_{<t}^i)$ (this is also the representation function $q(\mathbf{x})$); a prior network φ_τ^{prior}, which output parameters define the prior distribution $p(z_t^i|x_{<t}^i, z_{<t}^i)$; a decoder network φ_τ^{dec}, which output parameters define the generative model $p(x_t^i|z_{\leq t}^i, x_{\leq t}^i)$; feature extraction networks φ_τ^x and φ_τ^z for x and z, respectively; a recurrent process layer for hidden state update; a hypothesis function $\phi(z)$; and a function $h(\mathbf{x}_c)$ that maps causal features to the output intention. Specifically, all φ_τ consist of two hidden layers of 16 neurons with ReLU activation, where the input dimension is 4 and output dimension is 16. The hidden size of the recurrent layer is set to 16, where the input dimension is 32 and output dimension is 16. We consider ϕ to be the last layer of the network with one hidden layer of 16 neurons and ReLU activation, where the input dimension is 16 and output dimension is 2. We take h to be identity function for simplicity. The dimension of latent representation z is set to 2. When implementing state-of-the-art methods, since most of the baseline models are not designed for time-series input data, we add a RNN layer in front of each model instead of simply flattening the time-series input. The total number of parameters of all the baselines and our method are set to about equal.

E Additional Experiments and Results

E.1 Ablation Study

We conduct an extensive study using both FT and ZS data to investigate contribution of three components to our method's performance: loss for extracting temporal latent dependencies, contrastive loss for learning matching function, and metrics to evaluate distance between two representations. To be more specific, CTSDG w/o $\mathcal{L}_v$ is the CTSDG model without adding the loss term $\mathcal{L}_v$, where no temporal latent dependencies are modeled while learning invariant representations from the input data; CTSDG w/o ℓ_{con} is the CTSDG model without learning the matching function Ω, where the loss $\mathcal{L}_r$ for satisfying invariance condition of X_C is optimized by randomly selected input pairs; CTSDG w/ s_{ℓ_1} and CTSDG w/ s_{ℓ_2} are CTSDG models but using ℓ_1 and ℓ_2 as distance metric between two representations, respectively. From the ablation study results in Table 3, we observe that by capturing the temporal latent dependencies and learning matching function for input pairs that share causal features, the model is able to have better performance especially when target domain has large domain shift compared to source domains (the fourth row). Moreover, according to the results from using ℓ_1 and ℓ_2 as distance metric, the cosine similarity used in our method seems to be the most suitable measure in the causal feature space for this prediction problem.

E.2 Analysis on Invariant Representation

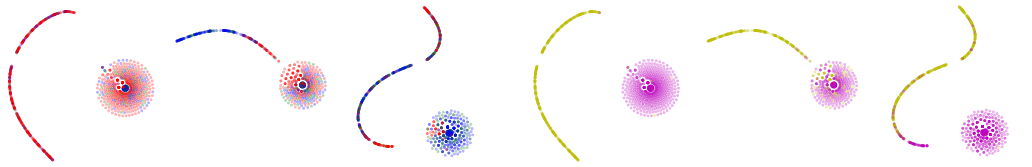
We utilize t-SNE [45] to analyze the quality of latent representation learned by our proposed model in Figure 5. We train our model using two domains (FT-1,2) and evaluate on two unseen domains (FT-3 & ZS). It appears from Fig. 5(a) and (b) that CTSDG achieved a higher overlap between train and test domains for FT-3 than ZS, highlighting the difficulty of generalizing to the ZS domain. Such finding coincides with our prior domain knowledge as well as quantitative results in Table 1. Moreover, based on Fig. 5(d) and (e), we find that the classes are well-separated in the training domains for both models. We also observe that when the intention label is *yield*, the data is more evenly dispersed. The

Table 3: Ablation study of our method with prediction accuracy (%).

Source	Target	CTSDG w/o $\mathcal{L}_v$	CTSDG w/o ℓ_{con}	CTSDG w/ s_{ℓ_1}	CTSDG w/ s_{ℓ_2}	Ours
FT-1,2	FT-3	75.79 (0.64)	83.24 (3.87)	81.75 (5.73)	79.7 (3.18)	86.03 (1.48)
FT-1,3	FT-2	95.83 (0.39)	95.15 (0.26)	95.91 (1.28)	96.08 (0.82)	96.85 (0.15)
FT-2,3	FT-1	97.49 (0.63)	97.45 (1.26)	96.2 (1.90)	95.19 (1.49)	98.85 (0.13)
FT-1,2,3	ZS	77.29 (1.82)	78.74 (3.02)	80.67 (5.54)	83.27 (4.74)	85.51 (2.66)
Average		86.60	88.65	88.69	88.56	**91.81**

reason could because the input time-series data with *yield* label are easier for the model to capture underlying domain representations that are invariant across time. Whereas the time-series data with *pass* label could have more temporally diversed domain-invariant representations.

In order to illustrate the effectiveness of learning match function Ω, we further utilize t-SNE to compare the latent representation learned by CTSDG with that by CTSDG without the contrastive loss in Eq. 3. Both models are trained using two domains (FT-1,2) and evaluated on an unseen domain FT-3. According to Fig. 5 (a) and (c), the model with learned match function has higher overlap between train and test domains than the model with random match function, especially when ground-truth target intentions are *yield*. We can also observe from Fig. 5 (d) and (f) that without minimizing contrastive loss (Eq. 3), target data from different classes cannot be well-separated in the latent representation, which generates miss-classification cases. Therefore, we can conclude that by enforcing the difference in individual representations for same-class inputs to be low through (Eq. 3), the model can better capture domain-invariant representations.



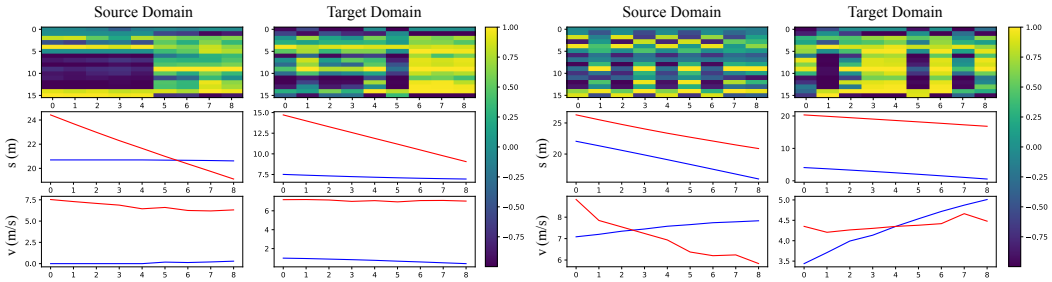
(a) FT-3 (b) ZS (c) FT-3 w/o ℓ_{con} (d) FT-3 (e) ZS (f) FT-3 w/o ℓ_{con}

Figure 5: The t-SNE plots for visualizing latent representations. The source domains come from FT-1 and FT-2. The target domain for (a)(c)(d)(f) and (b)(e) is from FT-3 and ZS, respectively. For better visualization, we color (a)-(c) based on domain information: blue dots for FT-1, green dots for FT-2, and red dots for corresponding target domain. In (d)-(f), we color data points by ground-truth vehicle intention labels, where yellow dots represent *pass* and magenta dots represent *yield*.

E.3 Analysis on Learned Temporal Latent Dependencies.

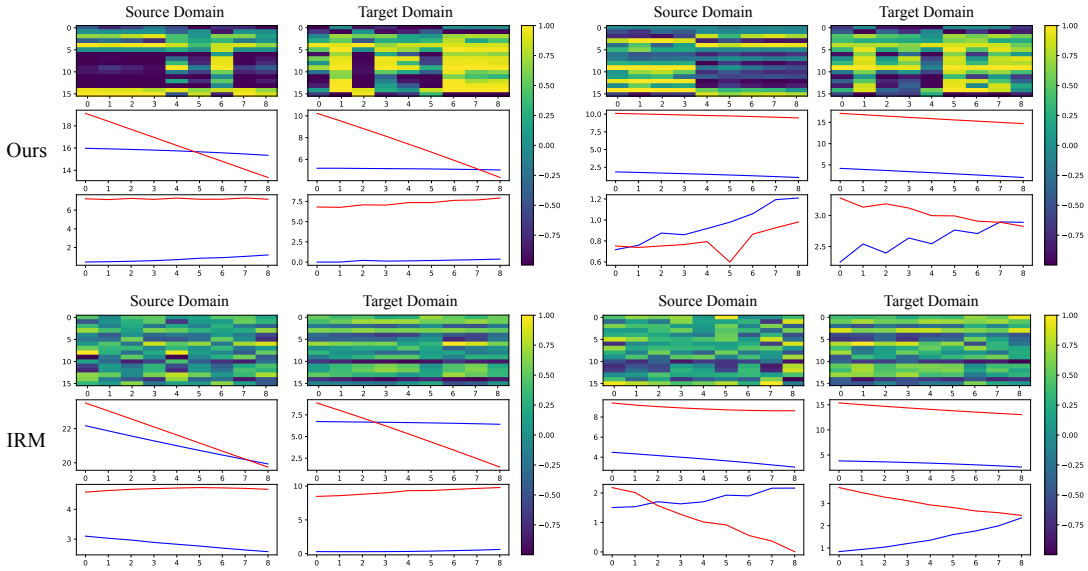
Figure 6 shows the temporal latent dependencies learned by our CTSDG for a given source-target pair. We first look at the two cases from source domain (i.e. first column in Fig. 6(a,b)). Looking at the neuron activation patterns along the time axis, we can see that neurons are more active when there are changes in either position or velocity profiles for two interacting vehicles. In fact, these changes are related to causal features which play important roles in determining the output intentions based on our proposed causal model. As we look at the two cases from target domain (i.e. second column in Fig. 6(a,b)), we observe consistent excitatory neuron firing patterns at critical time steps, which indicate that CTSDG successfully transfers knowledge to unseen domains. Therefore, we can conclude that our CTSDG method is able to capture temporal latent dependencies that are domain invariant and related to causal features, which enables it generalize time-series driving data to unseen domains.

According to Table 1, IRM has the best performance among all baseline methods when the source domain is FT-1,2 and target domain is FT-3. Therefore we compare the temporal latent dependencies learned by IRM with those learned by our method (see Figure 7). We observe that IRM results in sporadic activation pattern and fails to capture and transfer temporal latent dependencies. In contrast, our method can extract critical temporal dependencies that are relevant towards two vehicles' intention and such dependencies can be further transferred from source to target domain.



(a) intention = yield (b) intention = pass

Figure 6: Learned temporal latent dependencies across time for different output intentions. The source domains are FT-1,2 and the target domain is FT-3. Each column shows a single sample from either source or target domain. The top row represents the cell state of memory cell for CTSDG, where y-axis refers to the activation of each neuron and x-axis refers to activation per time-step. The second row plots two vehicles' longitudinal coordinates under the Frenét Frame with respect to time. The third row plots the velocity profiles for the two interacting vehicles.



(a) intention = yield (b) intention = pass

Figure 7: Learned temporal latent dependencies across time for different output intentions using different methods. The source domains are FT-1,2 and the target domain is FT-3. Each column under each method shows a single sample from either source or target domain. The top row under each method represents the cell state of memory cell, where y-axis refers to the activation of each neuron and x-axis refers to activation per time-step. The second row under each method plots two vehicles' longitudinal coordinates under the Frenét Frame with respect to time. The third row under each method plots the velocity profiles for the two interacting vehicles.

F Extension to Trajectory Prediction Tasks

In this paper, although we illustrate the performance of our proposed domain generalization method for vehicle intention prediction tasks, our CTSDG approach can also be directly or indirectly utilized for trajectory prediction tasks. The most straight forward way is to frame the trajectory prediction problem as classification over a diverse set of trajectories. In fact, according to [46], by framing trajectory prediction problem as classification over a diverse set of trajectories, we were able to a) ensure a desired level of coverage of the state space, b) eliminate dynamically infeasible trajectories, and c) avoid the issue of mode collapse.

Moreover, the accuracy of intention classification plays an important role in regression tasks. As stated and showed in [6, 34, 35, 36], goal/intent-conditioned trajectory forecasting can improve joint-agent and per-agent predictions, compared to unconditional forecast. Conditional forecasting can be also used for planning tasks, which effectiveness is demonstrated in [8]. Therefore, having an accurate and domain generalizable intention/goal predictor is the prerequisite of developing state-of-the-art trajectory predictor and motion planner.