CaDeT: a Causal Disentanglement Approach for Robust Trajectory Prediction in Autonomous Driving

Supplementary Material

1. Uncertainty-driven Causal Intervention

As mentioned in the paper, the proposed uncertainty-driven causal intervention method is designed to generate multiple distributions to simulate potential shifts during inference time. Here, we summarize the steps of the proposed method as described in Algorithm 1.

In this method, we first calculate uncertain feature statistics, mean and standard deviation, through the embedding of spurious patterns within a mini-batch. This process captures their basic statistical properties (a). We then proceed to estimate the uncertainty of these statistics to set the stage for simulating potential shifts that may arise due to perturbations during inference time (b).

We continue by synthesizing new feature statistics by random sampling from Gaussian distributions. These distributions are parameterized by the original statistics and their estimated uncertainties, introducing a probabilistic component into the latent space and facilitating the exploration of various distribution shift scenarios (c). Finally, to form the intervention set, we adjust the original embeddings using the newly intervened statistics (d).

Note that the proposed intervention procedure benefits from a differentiable sampling operation as in [4], making the module trainable to better simulate potential distribution shifts.

Algorithm 1: Causal Uncertainty-driven Intervention

Input : Embedding of spurious patterns $\mathbf{Z}_S$ (referred to as $\mathbf{Z}$), intervention set size n_I

Output : Intervention set I

a. Calculate uncertain feature statistics $\mu(\mathbf{Z}), \sigma^2(\mathbf{Z})$

b. Uncertainty estimation on the statistics

$\Sigma^2_\mu(\mathbf{Z}) = \frac{1}{b}\sum_{j=1}^{b}\left(\mu(\mathbf{Z}) - \mathbb{E}_j[\mu(\mathbf{Z})]\right)^2$

$\Sigma^2_\sigma(\mathbf{Z}) = \frac{1}{b}\sum_{j=1}^{b}\left(\sigma(\mathbf{Z}) - \mathbb{E}_j[\sigma(\mathbf{Z})]\right)^2$

// Create intervention set

$I = [\,]$

for $i = 0$ *to* $n_I - 1$ **do**

  c. Intervene feature statistics by sampling from the given Gaussian distributions

  $\beta_i \sim \mathcal{N}(\mathbf{0}, \mathbf{1}), \quad \gamma_i \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$

  $\mu(\mathbf{Z})^*_i = \mu(\mathbf{Z}) + \beta_i \Sigma_\mu(\mathbf{Z})$

  $\sigma(\mathbf{Z})^*_i = \sigma(\mathbf{Z}) + \gamma_i \Sigma_\sigma(\mathbf{Z})$

  d. Intervene embeddings of spurious patterns

  $I_i = \sigma(\mathbf{Z})^*_i \times \frac{\mathbf{Z} - \mu(\mathbf{Z})}{\sigma(\mathbf{Z})} + \mu(\mathbf{Z})^*_i$

  $I \leftarrow I_i$

end

return *Intervention set* I

Table 1. Comparison of training time for each batch on the Argoverse dataset using different variations of the proposed method.

Model	b-FDE$_6$	minADE$_6$	minFDE$_6$	MR$_6$	Training-time (ms)
Baseline	2.07	0.93	1.57	0.21	183
Baseline+Dis	2.03	0.88	1.48	0.20	188
CaDeT	**1.87**	**0.71**	**1.22**	**0.16**	201

2. Relative Temporal Encoding

As discussed in the paper, we use relative temporal encoding as in [2] to model the temporal dynamics in the DyHIN. This approach, inspired by positional encoding in Transformer models [8, 12], encodes time stamps into a sequence of sinusoidal functions. The key advantage of this method is its ability to capture the continuity and differentiability of time, essential for gradient-based optimization in GNNs.

Here, given a source node u and a target node v, along with their corresponding timestamps $t(u)$ and $t(v)$, we denote the relative time gap $\Delta t_{v,u} = t(v) - t(u)$ as an index to get a relative temporal encoding $R(\Delta t_{v,u})$. Then, a fixed set of sinusoid functions as the basis is used, followed by a learnable transformation layer ϕ to find the relative temporal encoding R. As shown below, for even indices $2i$ and odd indices $2i+1$ sine and cosine functions are used,

$$\begin{aligned} \text{Base}(\Delta t_{v,u}, 2i) &= \sin\left(\Delta t_{v,u}/10000^{\frac{2i}{d}}\right) \\ \text{Base}(\Delta t_{v,u}, 2i+1) &= \cos\left(\Delta t_{v,u}/10000^{\frac{2i+1}{d}}\right) \\ \text{R}(\Delta t_{v,u}) &= \phi\left(\text{Base}\left(\Delta t_{v,u}\right)\right) \end{aligned} \tag{1}$$

where d denotes the representation dimension. We use a 1-layer MLP as the transformation layer ϕ (refer to [2] for more details). Building on this approach, we concatenate the relative temporal encoding with the source node features, as outlined in Eq. 4 in the paper. This allows the GNN to incorporate temporal information when determining the importance or relevance of different nodes and their connections at various time steps.

3. CaDeT as a Plugging and its Training Cost

One advantage of the proposed method is that it can be incorporated in many trajectory prediction models as a plug-in. More specifically, the causal disentanglement can be used to separate the targeted representation into disjoint sets, including causal and spurious, and therefore utilize the proposed intervention combined with the invariance objective optimization to enable the prediction model to focus on causal factors, thus minimizing the negative effect of spurious correlations.

Adding a plug-in module to the model can add to the computational complexity. In what follows, we would examine the

Table 2. Quantitative results on the Waymo motion forecasting leaderboard. The "†" indicates an ensemble version. The "*" refers to predicting more futures than required. For each metric, the best result is in **bold** and the second best result is underlined.

		All agents				Vehicle			
Method	**Reference**	**mAP**	**minADE**	**minFDE**	**MR**	**mAP**	**minADE**	**minFDE**	**MR**
HDGT [3]	TPAMI 2023	0.283	0.570	1.143	0.144	0.324	0.668	1.347	0.140
SceneTransformer [6]	ICLR 2022	0.279	0.612	1.212	0.156	0.327	0.709	1.412	0.148
MTR* [9]	NeurIPS 2022	0.413	0.605	1.221	0.135	0.449	0.764	1.526	0.151
MTR++* [10]	Arxiv 2023	**0.433**	0.590	1.194	**0.130**	**0.487**	0.718	1.432	0.137
MultiPath++* [11]	ICRA 2022	0.409	0.556	1.158	0.134	0.463	**0.650**	1.355	**0.130**
WayFormer†* [5]	ICRA 2023	0.419	0.545	1.128	0.123	0.466	0.639	1.321	0.117
MTR++†* [10]	Arxiv 2023	0.463	0.558	1.117	0.112	0.514	0.668	1.317	0.114
MotionLM†* [7]	ICCV 2023	0.436	0.551	1.120	0.106	0.478	0.664	1.353	0.107
CaDeT (Ours)	-	0.390	**0.545**	**1.136**	0.140	0.449	0.651	**1.310**	0.132

computational overhead of our approach.

We model the representations based on spatiotemporal patterns. This design choice is motivated by the fact that causal factors in driving scenes occur simultaneously through spatial and temporal relations. As a result, we leverage a GNN as our predictor to encode scene representations. In this regard, we compare our training cost with that of a standalone GNN. We report our training time per batch across different variations of the model, including baseline (a standalone GNN), baseline+Dis (the GNN equipped with disentangled attention block), and CaDeT (our full model). All the experiments are conducted on an NVIDIA Tesla V100.

As shown in in Table 1, our disentangled attention block adds a training overhead as low as 2.7% and the proposed intervention coupled with the invariance objective termed CaDeT, imposes less that 10% overhead on the training. However, by introducing such a small overhead, we achieve significant improvement on all metrics. Particularly, baseline+Dis and CaDeT improve upon the baseline model on minFDE metric by $5.73\%(1.57 \rightarrow 1.48)$ and $22.29\%(1.57 \rightarrow 1.22)$, respectively. It is also worth noting that the overheads of the proposed intervention and the invariance optimization are only present during training stage, leaving only the overhead of the disentangled attention block during inference.

4. Comparison to State-of-the-art

We further evaluate our method against state-of-the-art models on the WOMD dataset, as shown in Table 2. We present the prediction results averaged over all agents and vehicles which we apply our interventions to. The models are categorized based on their additional processing overheads, including the models that make a large number of predictions by, e.g. oversampling, (indicated by "*") and the models that consists of the ensemble of several models (indicated by "†").

According to Table 2, as expected, the ensemble models generally outperform single models across most metrics, highlighting the effectiveness of ensemble techniques in enhancing prediction accuracy. However, aggregating several models can significantly increase complexity, making them less practical for real-world applications. Compared to single models, which is our case, despite focusing on improving robustness and generalizability against potential perturbations, our model's performance is comparable to the best models, and in some cases surpasses them on some metrics.

In vehicle prediction, our model, which has $\sim$ 7.8M parameters, performs better compared to single models without oversampling, namely HDGT ($\sim$ 12M) and SceneTransformer ($\sim$ 15M) across all metrics. Compared to other single models, our model is at very close second to Multipath++ on minADE and MR, and outperforms heavily parameterized MTR ($\sim$ 65M) on all metrics (by $14.79\%(0.764 \rightarrow 0.651)$ and $14.15\%(1.526 \rightarrow 1.310)$) and its variation MTR++ ($\sim$ 125M) on most metrics by up to 9%. Lastly, on minFDE, our model achieves the best performance overall, even in comparison to ensemble approaches. These results further confirm that our method is capable of accurately predicting behaviors, as represented in the final trajectory points. On the all agents category, as presented in the table, our method achieves overall second best performance on two metrics, while surpassing single models with no oversampling on most metrics and perform by par with others.

5. DRL in other domains.

DRL is more of a concept than a fixed method, focusing on learning data representations where individual factors or features are separated or disentangled. The key to DRL is how these representations are modeled and the underlying intuition of disentanglement, which shapes the objective function. Therefore, the design choices in DRL are tailored to the applications. In our work, inspired by the causal relationships in variable driving scenarios, we aim to disentangle causal from spurious patterns that result from vehicle interactions through time. Consequently, we defined our objective based on causal invariance theory. The closest work in other domains is [1], proposing a causal framework for representation learning that addresses disentanglement and the ability to withstand domain shifts.

References

[1] Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *ICML*, 2019. 2

[2] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Het-

erogeneous graph transformer. In *The World Wide Web Conference*, 2020. 1

[3] Xiaosong Jia, Penghao Wu, Li Chen, Hongyang Li, Yu Liu, and Junchi Yan. Hdgt: Heterogeneous driving graph transformer for multi-agent trajectory prediction via scene encoding. *arXiv:2205.09753*, 2022. 2

[4] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv:1312.6114*, 2013. 1

[5] Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S. Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. In *ICRA*, 2023. 2

[6] Jiquan Ngiam, Vijay Vasudevan, Benjamin Caine, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, et al. Scene Transformer: A unified architecture for predicting future trajectories of multiple agents. In *ICLR*, 2022. 2

[7] Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S. Refaat, Rami Al-Rfou, and Benjamin Sapp. Motionlm: Multi-agent motion forecasting as language modeling. In *ICCV*, 2023. 2

[8] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv:1803.02155*, 2018. 1

[9] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Mtr: Motion transformer with global intention localization and local movement refinement. *NeurIPS*, 2022. 2

[10] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Mtr++: Multi-agent motion prediction with symmetric scene modeling and guided intention querying. *arXiv:2306.17770*, 2023. 2

[11] Balakrishnan Varadarajan, Ahmed Hefny, Avikalp Srivastava, Khaled S Refaat, Nigamaa Nayakanti, Andre Cornman, Kan Chen, Bertrand Douillard, Chi Pang Lam, Dragomir Anguelov, et al. MultiPath++: Efficient information fusion and trajectory aggregation for behavior prediction. In *ICRA*, 2022. 2

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 1