25%	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
AutoBots [22]	0.942/1.886	0.333/0.662	0.563/1.156	0.446/0.958	0.409/0.904	0.539/1.113
+ Vanilla [38]	0.956/1.893	0.352/0.698	**0.537/1.143**	0.439/0.969	0.426/0.935	0.542/1.128
+ Augment [56]	0.942/1.867	0.359/0.722	0.544/1.152	0.441/0.971	0.409/0.922	0.539/1.127
+ Contrast (ours)	0.938/1.885	0.320/0.616	0.565/1.187	0.450/0.971	0.386/0.821	0.532/1.096
+ Ranking (ours)	**0.906/1.814**	**0.307/0.582**	0.556/1.157	**0.435/0.942**	**0.376/0.819**	**0.516/1.063**

Table 3. Quantitative results of sim-to-real transfer from ORCA simulations to 25% of the ETH-UCY dataset.

50%	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
AutoBots [22]	0.940/1.883	0.326/0.612	0.566/1.197	0.434/0.945	0.358/0.787	0.525/1.085
+ Vanilla [38]	0.923/1.913	0.342/0.661	**0.535/1.141**	0.430/0.954	0.383/0.886	0.523/1.111
+ Augment [56]	0.937/1.885	0.340/0.660	0.535/1.146	0.424/0.938	0.377/0.878	0.523/1.101
+ Contrast (ours)	0.935/1.870	0.344/0.667	0.554/1.148	0.422/0.913	0.346/0.772	0.520/1.074
+ Ranking (ours)	**0.903/1.810**	**0.300/0.566**	0.543/1.146	**0.420/0.911**	**0.331/0.725**	**0.499/1.029**

Table 4. Quantitative results of sim-to-real transfer from ORCA simulations to 50% of the ETH-UCY dataset.

100%	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
AutoBots [22]	0.938/1.916	0.334/0.678	0.550/1.152	0.420/0.916	0.343/0.787	0.517/1.090
+ Vanilla [38]	0.923/1.913	0.331/0.638	0.527/1.133	0.410/0.907	0.346/0.783	0.507/1.075
+ Augment [56]	0.938/1.878	0.332/0.635	**0.518/1.141**	**0.403/0.890**	0.348/0.796	0.508/1.068
+ Contrast (ours)	0.930/1.896	0.321/0.605	0.538/1.154	0.412/0.899	0.345/0.791	0.509/1.069
+ Ranking (ours)	**0.900/1.780**	**0.296/0.556**	0.539/1.148	0.408/0.889	**0.325/0.715**	**0.494/1.018**

Table 5. Quantitative results of sim-to-real transfer from ORCA simulations to 100% of the ETH-UCY dataset.

A. Additional Results

A.0.1. AutoBots on ETH-UCY

In addition to the aggregated results presented in Fig. 8, we summarize an in-depth breakdown of the sim-to-real transfer results in Tabs. 3 to 5. Across all evaluated settings, our ranking-based causal transfer consistently achieves superior prediction accuracy compared to the AutoBots baseline. Notably, it outpaces the standard sim-to-real method [38] in four out of the five subsets, with the sole exception in the UNIV subset. We conjecture that this exception might be attributed to the high similarity between the ORCA simulation and UNIV dataset.

To assess the stability of the ranking-based method, we conducted experiments using three different random seeds, as illustrated in Figs. 11 and 12. The results show that the ranking-based method consistently outperforms the vanilla model across all subsets of the ETH-UCY dataset. Furthermore, the ranking-based method typically yields smaller standard deviations, indicating more consistent performance.

Furthermore, as summarized in Tab. 6, AutoBots stands as one of the state-of-the-art models for multi-agent trajectory forecasting, leaving only marginal room for improvement on the ETH-UCY dataset. In spite of this, our proposed causal transfer method still offers notable improvements, resulting in more enhanced predictions. Please also note that this improvement is achieved through ORCA, while ORCA's performance on ETH-UCY dataset is very poor, as shown in Tab. 6. This marks the effectiveness of our methodology in extracting and transferring causal knowledge from a simulator, despite its big gap to the real world data.

Finally, we summarize the results of different methods in terms of Final Displacement Error (FDE) on the OOD test sets in Fig. 9. Similar to Fig. 6, our ranking-based method leads to the lowest prediction errors compared to the other counterparts, reaffirming its strength for boosting out-of-distribution robustness.

Deterministic	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
ORCA [65]	2.27/3.44	1.03/1.54	1.29/2.079	0.97/1.60	0.87/1.45	1.28/2.02
S-LSTM [2]	1.09/2.35	0.79/1.76	0.67/1.40	0.47/1.00	0.56/1.17	0.72/1.54
D-LSTM [32]	1.05/2.10	0.46/0.93	0.57/1.25	0.40/0.90	0.37/0.89	0.57/1.21
Trajectron++ [61]	1.02/2.00	0.33/0.62	0.53/1.19	0.44/0.99	0.32/0.73	0.53/1.11
Social-Transmotion[59]	0.93/1.81	0.32/0.60	0.54/1.16	0.42/0.90	0.32/0.70	0.51/1.03
AutoBots [22]	0.93/1.87	0.32/0.65	0.54/1.15	0.42/0.91	0.34/0.77	0.51/1.07
AutoBots + Ranking (ours)	**0.90/1.78**	**0.30/0.56**	0.53/1.12	0.41/0.89	0.32/0.70	**0.49/1.01**

Table 6. Comparison between different multi-agent forecasting models on the ETH-UCY dataset. Boosted by our proposed ranking-based causal transfer, the best result of AutoBots across three random seeds reaches comparable performance to the current state-of-the-art.

Dataset	ETH-UCY	NBA
AutoBots + Intervention [17]	0.53	0.58
AutoBots + Ranking (ours)	**0.49**	**0.54**

Table 7. **Comparison with causal intervention using AutoBot.** The ADE on ETH-UCY and NBA verify our causal transfer is robust under domain shifts across different datasets.

Model	K=1	K=5
Multi-Transmotion	0.379	0.272
Multi-Transmotion + Ranking (ours)	**0.342**	**0.264**

Table 8. **ADE of causal transfer from ORCA simulations to the JRDB dataset.** We outperform the Muti-Transmotion baseline by integrating our causal regularizers, showing the effectiveness of our approach in deterministic (K=1) and multimodal (K=5) setups. The numbers are in meters.

A.0.2. Comparison with other robust representations

We compare our ranking regularizer with causal intervention [9]. The results, detailed in Tab. 7, demonstrate that our method effectively mitigates the domain shift between simulated and real environments and better enhances performance compared to the proposed causal intervention.

A.0.3. Muti-Transmotion on JRDB

We further verify our causal transfer on a new model and dataset. We adopt Multi-Transmotion [18], a SOTA human motion prediction model, on JRDB[19], a social navigation dataset. We split an indoor and an outdoor scene as the testing set, and the other as the training set. The results in Tab. 8 validate the effectiveness of our causal regularize, which consistently improves the model, to demonstrate the proposed ranking regularizer can be implemented in various baseline scene representations and has the potential to improve them in many settings. In addition, the second column in Tab. 8 shares our improvement in the multimodal setup. In this multi-task approach, we learn the multimodal trajectory prediction task on the real-world dataset, while at the same time we regularize the representation learned by the encoder using our causal ranking regularizer, on the synthetic data.

B. Implementation Details

Experiment details. Our experiments are largely built upon the public code of prior work, with as few modifications as possible made for the implementations of our proposed regularizers. Concretely, in the robustness analysis reported in Tab. 1, we train each model on our constructed dataset using the default hyperparameters of the corresponding baseline. To understand the performance of our proposed methods in Sec. 5.2, we fine-tune the pre-trained checkpoint for 10 epochs, and evaluate the obtained model on the hold-out test set. The main hyperparameters used for training our baseline model AutoBots [22] and the causal regularizers are listed in Tab. 9.

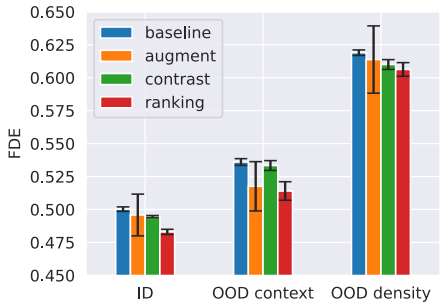


Figure 9. Additional quantitative results of our method on the out-of-distribution test sets, as a supplement to Fig. 6. Models trained by our method yield lower FDE. Results are averaged over five random seeds.

Simulation details. Our diagnostic dataset is generated using a customized version of the Reciprocal Velocity Obstacle simulator that employs Optimal Reciprocal Collision Avoidance (ORCA) [66]. To simulate realistic causal relationships between agents, we imposed a visibility constraint where an agent only observes other neighbors within their proximity and its 210° field of view. This visibility plays a significant role in determining the influence of one agent on another. Specifically, we define a neighbor i as having a *direct influence* on the ego agent at time step t if it is visible to the ego in that time step, *i.e.*, $\mathbb{1}_t^i = 1$. Additionally, we introduce a visibility window that records agents that were previously visible, facilitating the modeling of a richer spectrum of direct and indirect inter-agent influences. To encourage the presence of non-causal agents in dense spaces, we explicitly directed specific agents to follow others, thereby making them non-causal or indirect causal.

Dataset details. Tab. 10 summarize the key statistics of our diagnostic datasets, including both the in-distribution (ID) training set and the out-of-distribution (OOD) test set. Specifically, we consider two distinct types of OOD datasets, each deviating from the ID dataset in specific aspects, such as agent density and/or scene context.

- *ID*: The training dataset is characterized by an average of 12 pedestrians per scene, each interacting with a few others to navigate towards their goals. All the scenes are set in an open area context, allowing unrestricted movements and serving as the base environment in our experiments.
- *OOD Density*: In our first OOD set, we retain the same context setting as the ID dataset but increase agent density. Specifically, we introduce more agents in proximity to the ego agent to intensify agent interactions. Additionally, we add agents behind the ego agent, which results in more non-causal agents. This dataset aims to test the robustness of the model in handling increased agent density.
- *OOD Context*: The second OOD set alters the scene context from an open area to a narrow street, where pedestrians walk from one end to the other. Given that agents walking in the same direction generally do not interact, we double the number of agents in the scene, thus ensuring a similar degree of interaction complexity to the ID dataset.

Exemplary animations for each data split can be found in our public repository.

Baseline details. To the best of our knowledge, few prior work studies causally-aware representation learning in the multi-agent context. To examine the efficacy of our proposed methods, we consider the following three existing methods as comparative baselines.

- *Baseline*: the baseline method trains the model on the data in the target domain only, *i.e.*, the AutoBots baseline trained on the in-distribution dataset in Sec. 5.2 or the real-world ETH-UCY dataset in Sec. 5.3.
- *Vanilla*: the vanilla sim-to-real method combines simulated and real-world data in the training process. It adheres to a standard prediction task, with an equal mix of data from each domain in every training batch [38].
- *Augment*: the causal augmentation method is built upon the *Baseline* for the experiment in Sec. 5.2 or the *Vanilla* for the experiment in Sec. 5.3. It augments training data by randomly dropping non-causal agents based on the provided annotations [56].

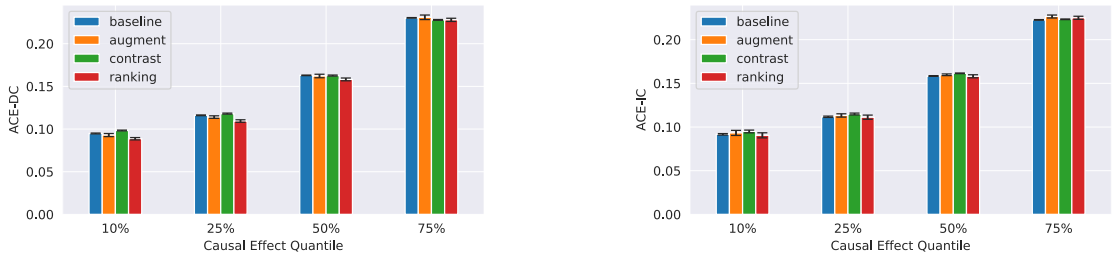
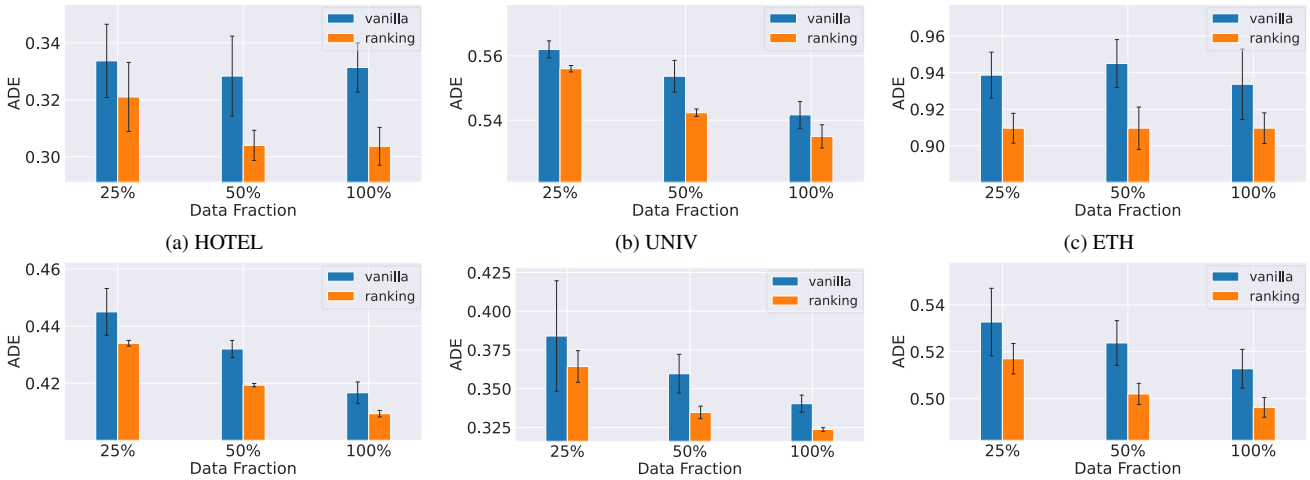


Figure 10. Additional quantitative results of our method on causal effect estimates, as a supplement to Fig. 6. Models trained by our method yield lower ACE-DC and ACE-ID, especially noticeable in scenarios with low-quantile ground-truth causal effects. Results are averaged over five random seeds.

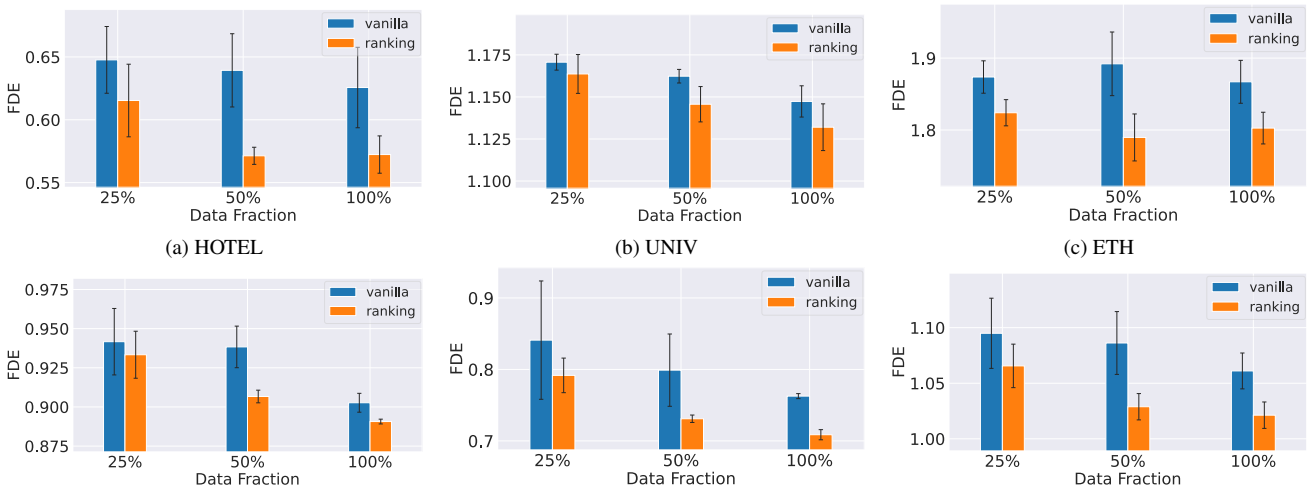
.



(a) HOTEL (b) UNIV (c) ETH

(d) ZARA1 (e) ZARA2 (f) Average

Figure 11. Additional results of our ranking-based causal transfer on the ETH-UCY dataset, as a supplement to Fig. 8. The results of ADE are averaged on each subset over three random seeds.



(a) HOTEL (b) UNIV (c) ETH

(d) ZARA1 (e) ZARA2 (f) Average

Figure 12. Additional results of our ranking-based causal transfer on the ETH-UCY dataset, as a supplement to Fig. 8. The results of FDE are averaged on each subset over three random seeds.

C. Additional Discussions

Counterfactual simulation. Our diagnostic dataset, enabled by counterfactual simulations, offers clean annotations of causal relationships, serving as a crucial step in understanding causally-aware representation of multi-agent interactions. However, the realism of these simulated causal effects is still subject to some inherent limitations. For example, we have enforced a stringent constraint on the field of view for each agent, considering that the ego agent is usually unaffected by trailing neighbors. Such constraints could compromise the optimality of the ORCA algorithm, potentially resulting in unnatural trajectories. We believe that integrating more advanced simulators, *e.g.*, CausalCity [47], can address these challenges and we anticipate promising outcomes along this line for future research.

Multi-agent causal effects. Our annotation and evaluation have been focused on the causal effect at an individual agent level, namely we remove only one agent at a time. This aligns with the notion of Causal Agents [56], *i.e.*, a neighboring agent has a certain causal relationship with the ego agent. Through this lens, we observe that while recent representations are already partially robust to non-causal agent removal, they tend to underestimate the effects of causal agents. However, it is important to note that this is still a rather simplified and restricted setting compared to the group-level causal effects, where the collective behavior of multiple agents may have a more complex influence on the ego agent. Understanding and addressing this challenge can be another exciting avenue for future research.

name	value
batch size	16
pre-training learning rate	7.5×10^{-4}
fine-tuning learning rate	2.34375×10^{-5}
contrastive weight α	1000
ranking weight α	1000
ranking margin m	0.001
non-causal threshold ϵ	0.02
causal threshold η	0.1

Table 9. Key hyper-parameters in our experiments.

Dataset	Number of scenes		Number of agents per scene			
	train	test	non-causal	direct causal	indirect causal	total
ID	20k	2k	1.31	8.35	0.48	13.03
OOD Context	-	2k	6.13	9.92	1.47	21.39
OOD Density	-	2k	9.47	12.27	2.55	29.98

Table 10. Key statistics of our diagnostic datasets.

D. Qualitative Examples of Synthetic Data

In this section we share some qualitative examples of our synthetic dataset, with the non-causal, directly causal, and indirect causal labels in them. Some scenarios are depicted in Fig. 13, where the points on the lines represent the position of agents in the last time step before the prediction horizon. Lonely dots without any lines on them represent static agents that do not move in the scene.

Figure 13. **Visualizations of our synthetic dataset with causal labels, generated using ORCA simulator.** The dots indicate the position of the agent in the last timestep before the prediction horizon, and single dots without any lines associated with them represent static agents who do not move in the scene.