Method	Vanilla (ERM)	Counterfactual	Invariant (ours)
ADE (↓)	0.536 ± 0.015	0.512 ± 0.057	**0.457** ± 0.054
FDE (↓)	1.088 ± 0.039	1.029 ± 0.136	**0.918** ± 0.098

Table 2. Comparison of different methods on the original ETH-UCY dataset. The STGAT [35] trained by our invariant approach substantially outperforms the vanilla ERM and the counterfactual counterparts [13]. The number of sampled trajectories is set to 1 for computational efficiency. Results are averaged over 5 seeds.

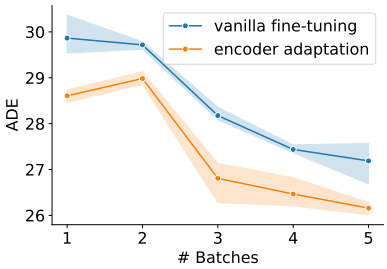


Figure 8. Quantitative results of low-shot transfer on the SDD [61] dataset. Our modular adaptation strategy yields higher sample efficiency than the conventional fine-tuning.

A. Additional Experiments

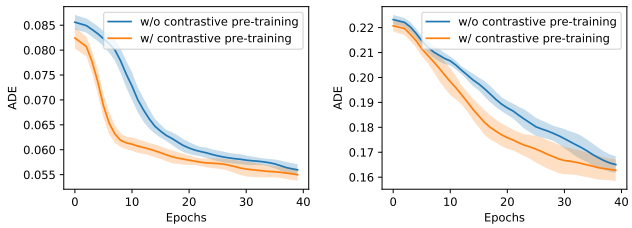
Robustness on the original ETH-UCY dataset. In addition to the robustness experiments under controlled distribution shifts of spurious features (§4.1) or style features (§4.2), we also evaluate our method on the original ETH-UCY dataset, where each subset is subject to some unknown selection biases. We train the STGAT [35] on four subsets ('eth','univ','zara1' and 'zara2') and test it on the rest ('hotel'). The results in Table 2 confirms the strength of our method on real-world data.

Low-shot transfer on the SDD dataset. Apart from simulated style shifts in §4.2, we further evaluate our method on the SDD dataset under substantial style shifts. We create four different domains according to the agent type and average speed. We use three domains for training and the last one for evaluation. We apply our method on top of the Y-Net [50], and compare our modular adaptation strategy against the standard fine-tuning of the entire model for low-shot transfer. The results in Fig. 8 demonstrate the scalability of our method to real-world style shifts.

Larger style shifts. As a supplement to Table 1, we summarize the detailed results under larger style shifts in Table 3. Among these OOD test domains ($d > 0.5$), the farther the style parameter is from the training ones, the larger improvement we obtain from using the full version of our method. This result confirms the advantage of our modular design with an enforced structure of the invariant and style knowledge for robust generalization.

Method	$d = 0.6$	$d = 0.7$	$d = 0.8$
Vanilla (ERM)	0.192 ± 0.013	0.246 ± 0.020	0.309 ± 0.025
Invariant (ours)	0.191 ± 0.007	0.245 ± 0.009	0.309 ± 0.011
Modular (ours)	0.112 ± 0.004	0.169 ± 0.011	0.242 ± 0.020
Inv + Mod (ours)	**0.107** ± 0.007	**0.156** ± 0.013	**0.221** ± 0.020

Table 3. ADE scores of different methods on OOD-Extra domains. The full version (invariant + modular) of our method yields more performance gains with an increasing degree of style shifts.



(a) IID style $d = 0.1$ (b) OOD style $d = 0.7$

Figure 9. Comparison of the models with and without style contrastive pre-training. The model pre-trained on the style contrastive task converges faster than the counterpart during the end-to-end training in both domains.

Style contrastive pre-training. As described in §3.4, one advantage of incorporating the proposed style contrastive loss is to ease the training of our modular model that consists of multiple sub-networks. In Figure 9 we compare the performance of the models during training with and without the style contrastive pre-training. The model pre-trained on the style contrastive task learns significantly faster than the counterpart without it.

B. Experimental Details

B.1. Spurious Shift Experiments

Experimental design. To clearly examine the robustness of motion forecasting models against spurious shifts (§4.1), we introduce an additional input variable σ_t, concatenated to the 2D coordinates. Its value is defined as a linear function of the trajectory curvature γ_t. By changing the scaling coefficient α, we artificially control a varied degree of spurious shifts. This controlled setting allows us to simulate the spurious correlation arising from two co-occurring phenomena in crowded spaces: observations become noisy due to occlusions; trajectories become non-linear because of interactions. Given this coincidence, statistical models may exploit the level of noise σ_t to ease predictions. Yet, such non-causal models are brittle. Any changes of the noise pattern (*e.g.*, due to perception algorithm updates illustrated in Figure 3) may degrade forecasting accuracy.

Architecture. For the experiments reported in §4.1, we use the standard STGAT [35] architecture for a fair comparison with the counterfactual analysis approach [13]. In order to take the x and y coordinate as well as the observa-

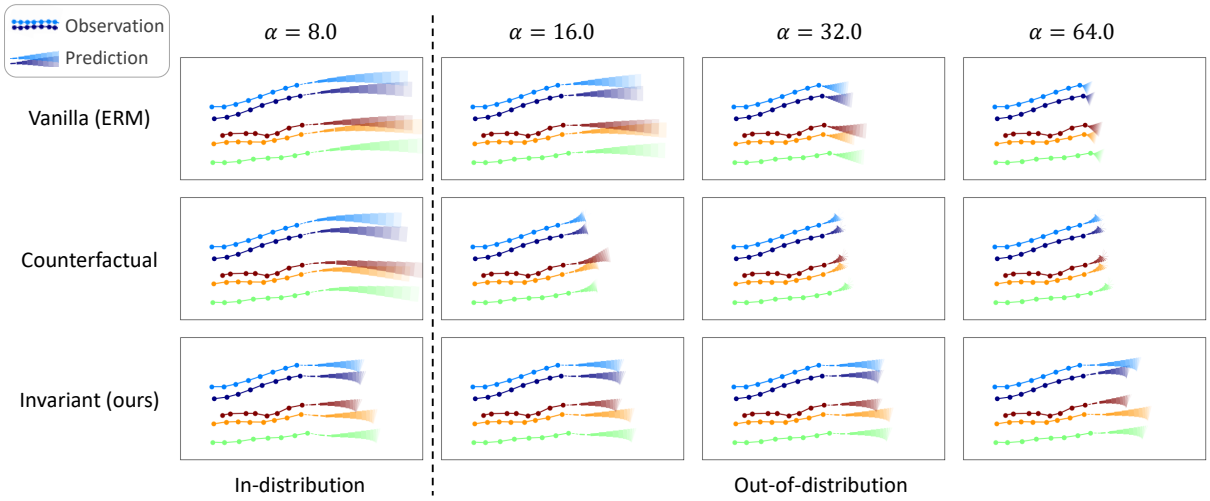


Figure 10. Visualization of the predicted trajectories from different methods in a particular test case of the ETH-UCY dataset with controlled spurious features. Despite the *same* past trajectory observation and ground truth future, the predicted trajectories from the two baselines abruptly slow down, when the strength of the spurious feature at test time is larger than that in the training domains. In comparison, our invariant learning approach results in much more robust solutions, even under substantial spurious shifts (*e.g.*, $\alpha = 64.0$).

tional uncertainty σ_t as inputs, we adjust the input dimension of the first LSTM module to three. All the remaining configurations align with the original STGAT. Following the previous work [13], we train the model in three steps: (i) pre-train the first LSTM, (ii) pre-train the GAT together with the second LSTM, (iii) train the whole model.

Hyper-parameters. We use the same hyper-parameters as in the original STGAT [35]. For the invariant penalty coefficient λ, we run grid search in a range from 0.001 to 100. Since the focus of our experiments is on the robustness and adaptability under distribution shifts rather than the performance on training domains, we only predict one trajectory output per instance instead of multiple ones [29] during training, which reduces computational expenses for the comparison of different methods. Other detailed hyper-parameters are summarized in Table 4.

B.2. Style Shift Experiments

Architecture. For the experiments in §4.2, we use a PECNet-like [51] feedforward network as our base model. Specifically, we model all components using MLPs. We train the modular network in four detailed steps: (i) train the invariant encoder together with the decoder, (ii) subsequently pre-train the style encoder and the projection head, (iii) followed by the style modulator, and (iv) finally train the entire model end-to-end.

Hyper-parameters. We keep most hyper-parameters identical to the setup in Appendix B.1. We further tune the learning rates for each module separately due to their distinct properties. Detailed settings for training, adaptation and refinement are summarized in Table 5.

config	value
batch size	64
epochs per stage	150, 100, 150
learning rate	0.001

Table 4. Hyper-parameters in spurious shift experiments.

config	value
batch size	64
epochs per stage	100, 50, 20, 300
contrastive loss coefficient	1.0
learning rate baseline	0.001
learning rate style encoder	0.0005
learning rate projection head	0.01
learning rate style modulator (train)	0.01
learning rate style modulator (adapt)	0.001

Table 5. Hyper-parameters in style shift experiments.

B.3. Other Details

We train all of our models on a single NVIDIA Tesla V100 GPU. Each run takes around one hour. The source code of our method as well as baselines can be found at https://github.com/vita-epfl/causalmotion and https://github.com/sherwinbahmani/ynet_adaptive.

C. Additional Discussions

To the best of our knowledge, our work provides the first attempt to incorporate causal invariance and structure into the design and learning of motion forecasting models. Despite encouraging results, our work is still subject to a couple of limitations.

Limitations & future work. One major technical limitation lies in the granularity of the considered causal rep-

resentations. While our method places great emphasis on three prominent groups of high-level latent features, we have largely overlooked the structure of fine-grained features. One interesting direction for future work is to further exploit detailed causal structure for motion forecasting, for instance, (i) disentangling the left or right-hand traffic rules from social distance conventions within the group of style confounders, (ii) encouraging sparse interplay between submodules, *e.g.*, pruning the connections between inertia features and left or right-hand traffic rules, given their presumably minute significance.

Another limitation of our work is tied to the scale and diversity of experiments. Thus far, we have demonstrated the strengths of our method on two human motion datasets and two base models as proofs of concept. Nevertheless, our method is highly generic and we hypothesize that it can also bring similar benefits to other types of motion problems and datasets, *e.g.*, vehicles [11], sports [84] and driving simulations [52]. Extending the current empirical findings to more contexts can be another valuable avenue for future work.

Societal impact. Out-of-distribution robustness remains a salient weakness of motion forecasting models while having a crucial impact on the safety of autonomous systems, especially in the context of autonomous driving. Even though these machines operate accurately in their training environments, deploying them in unseen test conditions can result in undesired behavior, which may ultimately lead to fatal consequences in specific scenarios. With our work, we contribute to reducing this performance gap. However, we are aware of the remaining deficits of our motion forecasting approach in significantly changing conditions that should not be neglected when utilizing such systems in real-world applications.