Unveiling Transferability in Trajectory Prediction via Latent Scene Embeddings

Theodor Westny[1], David Axelsson[1], Björn Olofsson[1,2], and Erik Frisk[1]

[1] Division of Vehicular Systems, Linköping University, Linköping, Sweden
{theodor.westny, david.axelsson, erik.frisk}@liu.se
[2] Department of Automatic Control, Lund University, Lund, Sweden
bjorn.olofsson@control.lth.se

Abstract. The growing availability of trajectory datasets has fueled major advances in data-driven motion prediction. Yet, models trained on one dataset often fail to generalize beyond their training domain as a result of differences in scene layouts, agent behaviors, and sensing conditions. A framework that learns latent representations of datasets and quantifies their similarity using distributional metrics is presented. This large-scale study covers 24 major datasets, including the most widely used motion-prediction benchmarks, and shows that the resulting transferability scores strongly correlate with cross-dataset model performance. The results provide practical guidance for dataset selection, pretraining, and large-scale foundation models for motion prediction, paving the way toward more generalizable and robust predictive systems.

1 Introduction

Trajectory prediction plays an important role in applications such as autonomous driving, robotics, and human behavior analysis [23, 45, 87]. It involves forecasting the future states of agents based on past observations and interactions with their surroundings. The topic has gained significant attention in recent years, driven by advances in deep-learning methods, increased availability of trajectory datasets [6, 8, 9, 11–13, 22, 42, 52, 53, 55, 56, 65, 71, 76, 85, 86, 96, 98, 103, 105], and open-source tools [25, 48, 58, 93]. However, this progress has also highlighted an important challenge: trajectory datasets differ significantly in terms of agent types, environmental complexity, and sensing modalities. This diversity offers valuable opportunities for comprehensive model development, but it also poses difficulties for transferring models across different domains. A key open question is how well a model trained on one dataset generalizes to another—and whether such transferability can be predicted.

Advances in large foundation models demonstrate that training on diverse datasets can yield highly transferable representations [10,29,87,107]. Theoretically, this could be applied to trajectory prediction by pooling all available data. However, several factors limit the practicality of this approach. First, real-time systems such as autonomous vehicles operate under strict computational and latency constraints, making it impractical to deploy models at the scale and with the inference costs of current foundation models. Furthermore, while model compression and distillation techniques alleviate these issues [24,36,40,41], recent work shows that more data are not always better; low-quality or irrelevant data can degrade performance [54,77,83,108]. These observations motivate prioritizing the most relevant datasets for pretraining, rather than indiscriminately aggregating all available data.

A natural first step toward addressing this challenge is to compare datasets based on their surface-level features [4, 48, 78]. These analyses can highlight differences in statistical properties, including agent densities, speed profiles, and map structures. However, they do not inherently capture emergent behavioral characteristics, such as interaction patterns, social norms, or task intent, which are defining features of multi-agent motion-prediction problems [82].

In this work, transferability is studied through a learned latent space that captures behavioral characteristics across datasets, as illustrated in Fig. 1. This space is obtained by jointly training a single embedding model on all datasets under investigation to project scenarios into a representation where distances reflect structural and behavioral patterns in addition to surface-level feature similarities. This learned representation provides a more meaningful basis for assessing dataset transferability by leveraging the capacity of deep models to capture complex interactions. Once trained, the embedding model enables querying at the dataset level, supporting data-driven recommendations of suitable source datasets for pretraining or adaptation. In practice, transferability is estimated using the Kullback–Leibler (KL) divergence between Gaussian approximations of the latent dataset distributions. It is demonstrated that smaller latent distribution divergences correspond to improved transfer performance.

1.1 Contributions

The primary contributions of this paper are:

- **Dataset-level representation learning for trajectory prediction.** A latent embedding framework is introduced, modeling trajectory datasets as distributions in a shared latent space to enable comparison through their distributional geometry.
- **A directional probabilistic measure of transferability.** A divergence-based transferability metric is proposed to capture asymmetric transfer success, demonstrating that latent distances predict both zero-shot and fine-tuned performance.
- **A systematic large-scale study of cross-dataset transfer.** A comprehensive evaluation across 24 trajectory datasets and 552 transfer pairs is conducted, revealing underutilized datasets that improve performance on popular motion-prediction benchmarks.

Implementations are available at `https://github.com/westny/transferatlas`.

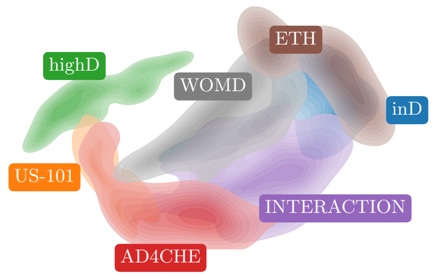


Fig. 1: t-SNE projection of learned scenario embeddings (contours indicate density). Dataset proximity (*e.g.*, ETH/inD, INTERACTION/WOMD) indicates potential for cross-dataset pretraining or knowledge transfer.

2 Related work

2.1 Trajectory prediction

Trajectory prediction remains an active area of research as a result of its central role in domains such as autonomous driving and robotics [23,45,87]. Advances in deep learning and increased data availability have made learning-based methods the dominant approach. A wide range of architectures have been explored, including recurrent networks [2,18,70,94], convolutional encoders [17,18,106], Transformer-based models [34,46,62,67], and graph neural networks (GNNs) [20, 32,57,59,80,89,95].

A substantial portion of the literature focuses on modeling interactivity, *i.e.*, how agents influence one another and how they are affected by their surrounding environment. This includes pooling-based methods [2,18,39,70,106], as well as approaches that explicitly model interactions using GNNs or Transformer-based architectures [20,32,57,67,80,95,102]. Environmental context is processed using specialized modules and usually incorporated through either rasterized scene encodings [17,57,68,69,74,80,106,109] or as vectorized maps [19,30,32,60,109].

Further, probabilistic modeling is widely used to capture the inherent uncertainty of future behavior. These models can be broadly categorized as *discriminative*, which directly predict multi-modal trajectory distributions [18,43,61,69,70, 95,109], or *generative*, which aim to learn the underlying distribution from which future trajectories can be sampled [3,5,15,38,39,50,57,67,79,80,92,97,102,106].

2.2 Domain adaptation and generalization

Improving model robustness across environments often involves either domain adaptation or domain generalization. Domain adaptation aims to reduce the discrepancy between the source and target data distributions, while domain generalization seeks models that perform well on unseen domains without additional tuning. In the field of trajectory prediction, several works explore these themes.

Recent approaches propose architectural adaptations such as ensemble-based methods [91], or specialized modules [88,104] to improve generalization. Other strategies tackle this problem by learning shared representations across datasets [5, 73, 81], or by designing dataset-agnostic input abstractions that reduce reliance on domain-specific cues [25, 33, 44, 48, 49, 100, 108].

On the domain adaptation side, recent work proposes techniques such as attention-based adaptation [99], feature alignment [31], and instance-level augmentations [51]. Other approaches adapt pre-trained models to new domains [84] or introduce meta-learning strategies for online or few-shot adaptation [47].

While domain adaptation and generalization are central themes in this work, the focus is not on developing new methods to improve these properties, but rather on understanding whether they can be predicted. Furthermore, evaluations are typically restricted to a limited set of source–target pairs. This leaves a clear gap for research that investigates transferability across a broader, more diverse spectrum of datasets.

2.3 Dataset similarity and transferability metrics

Quantifying dataset similarity is important for anticipating transfer learning success [1, 16, 21, 90, 101]. A closely related approach to ours is presented in [16], where methods to estimate transferability between time-series datasets are proposed. A shared autoencoder embeds source and target data into a common latent space, from which metrics such as the L1 distance between dataset centroids are derived and shown to correlate with transfer learning gains. In contrast, this work focuses on trajectory prediction, where spatio-temporal structure, interaction dynamics, and map-based context require specialized modeling [23, 45, 87]. In addition, probabilistic and non-symmetric measures, such as KL divergence, are considered to capture the inherently directional nature of transferability.

Related work on latent representations of trajectory datasets includes ScenarioNet [58], which utilizes TrafficGen [26] to obtain scenario embeddings that are then projected onto a lower-dimensional manifold. These projections highlight differences between synthetic and real data as well as residual variation across datasets. While primarily used for visualization, such an approach also highlights the potential of using latent space analyses to identify distributional gaps.

3 Problem definition

The problem of learning the approximate distributions of multiple trajectory datasets and using these distributions to quantify pairwise transferability is considered. Each dataset consists of interactive *scenes*, where a scene is defined as a set of agents (*e.g.*, vehicles, pedestrians) and their associated features (*e.g.*, position, velocity) that evolves over time. The temporal span of a scene is divided into a historical segment and a future segment. Each time instant in the scene is modeled as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of agents (nodes) and $\mathcal{E}$ is the set of edges representing the interactions between said agents. Let $N = |\mathcal{V}|$ be the number of agents, H the length of the historical segment, and F the length of the future segment. The node input features are summarized in a

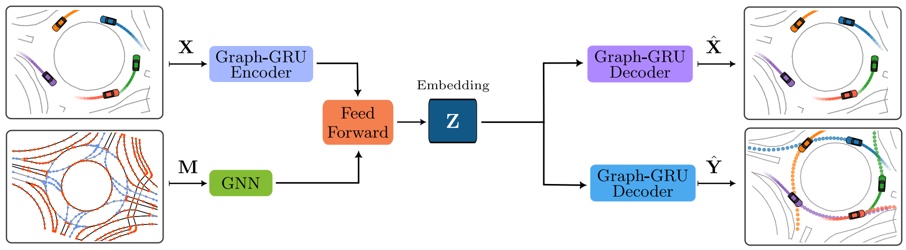


Fig. 2: Latent embedding model architecture. An encoder maps input features to node-level latent variables, which are aggregated into scene- and dataset-level embeddings. Two decoder heads handle feature reconstruction (predicting $\mathbf{X}$) and future state forecasting (predicting $\mathbf{Y}$). Training optimizes a combined reconstruction and forecasting loss.

tensor $\mathbf{X} \in \mathbb{R}^{N\times H\times D}$, and the output features in $\mathbf{Y} \in \mathbb{R}^{N\times F\times D}$, where $D = 5$, corresponding to 2D position, 2D velocity, and orientation. When available, the map $\mathbf{M}$ is also incorporated, encoded as a lane graph with associated features, shared across all time instants of the scene. For brevity, the notation is overloaded, letting $\mathbf{X}$ implicitly denote the pair $(\mathbf{X}, \mathbf{M})$ whenever map information is available.

Next, latent variables $\mathbf{z}_i$ are introduced for each node i, summarized in a matrix $\mathbf{Z} \in \mathbb{R}^{N\times L}$. Finally, for a scene $\mathcal{S}$, the latent scene embedding $\mathbf{Z}_{\mathcal{S}} \in \mathbb{R}^{L}$ is defined, and for a dataset $\mathcal{D}$, the corresponding latent dataset embedding $\mathbf{Z}_{\mathcal{D}} \in \mathbb{R}^{L}$, where L denotes the latent dimension. Unless otherwise stated, $L = 32$ in this work. The objective is two-fold:

1. Learn the latent dataset-level representations $\{\mathbf{Z}_{\mathcal{D}j}\}_{j=1}^{K}$ for datasets $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_K$ and use them to quantify pairwise transferability based on distributional (dis)similarity.
2. Empirically validate the inferred transferability by training a trajectory prediction model $\mathbf{X} \mapsto \mathbf{Y}$ on dataset $\mathcal{D}_i$ and evaluating it on a different dataset $\mathcal{D}_j$, for multiple pairs $(\mathcal{D}_i, \mathcal{D}_j)$.

4 Latent embedding model

To construct compact and comparable representations of the datasets under consideration, a deep latent embedding model is employed, designed to learn expressive codes of the individual samples (scenes) comprising each dataset.

To learn representations that capture all aspects of the data under a setting relevant to trajectory prediction, the model is trained using two complementary supervision signals: reconstruction of observed inputs and prediction of future states. The predictive objective reflects the trajectory forecasting setting, where past motion is observed and future motion is unknown, while reconstruction promotes information preservation. Latent regularization is used to encourage informative and stable representations [7, p. 564]. In particular, a normalization constraint is applied on the latent variables, $\|\mathbf{z}\|_2^2 = 1$ that restricts them to the surface of a unit sphere, encouraging comparability across samples and datasets while preventing degenerate solutions.

4.1 Model architecture

The latent embedding model follows an encoder–decoder architecture, illustrated in Fig. 2. The encoder maps input features to node-level latent variables, used to compute scene- and dataset-level embeddings. Two decoder heads are employed: one reconstructs input features, and the other predicts future states. The main building blocks of both networks are graph-gated recurrent units (Graph-GRUs) [95].

Graph-gated recurrent unit The Graph-GRU enables spatio-temporal interactions to be captured by replacing the linear mappings in the gated recurrent unit (GRU) [14] with GNN components. The original Graph-GRU design is simplified by using a minimal mechanism based on the minGRU [27], reducing model complexity with minor performance trade-offs.

At each time step k, a message-passing operation incorporates information from node i and its neighborhood $\mathbf{N}(i)$, producing intermediate vectors:

$$\left[\boldsymbol{v}_k^{(i)}, \boldsymbol{w}_k^{(i)}\right] = \mathrm{GNN}\left(\boldsymbol{x}_k^{(i)}, \{\boldsymbol{x}_k^{(j)}\}_{j\in\mathbf{N}(i)}\right), \tag{1}$$

where GNN$(\cdot)$ is the graph neural network operator from [72]. These are then used to update the hidden state $\boldsymbol{h}_k^{(i)}$ using the following gating mechanism:

$$\boldsymbol{z}_k^{(i)} = \sigma(\boldsymbol{v}_k^{(i)} + \boldsymbol{b}_z), \tag{2a}$$

$$\tilde{\boldsymbol{h}}_k^{(i)} = \tanh(\boldsymbol{w}_k^{(i)} + \boldsymbol{b}_h), \tag{2b}$$

$$\boldsymbol{h}_k^{(i)} = (1 - \boldsymbol{z}_k^{(i)}) \odot \boldsymbol{h}_{k-1}^{(i)} + \boldsymbol{z}_k^{(i)} \odot \tilde{\boldsymbol{h}}_k^{(i)}, \tag{2c}$$

where $\boldsymbol{b}_z$ and $\boldsymbol{b}_h$ are learnable bias terms, $\odot$ denotes the Hadamard product, and σ is the sigmoid function.

4.2 Training objective

The model is trained to minimize a combined objective of reconstruction and prediction losses. Node-level latent variables $\mathbf{z}_i$ are inferred from input features $\mathbf{X}_i$ using an encoder module Enc. To ensure the representation captures structural data properties and relevant forecasting features, the total loss $\mathcal{L}$ is defined as

$$\mathcal{L} = \|\,\mathsf{Rec}(\mathbf{Z}) - \mathbf{X}\|^2 + \|\,\mathsf{Pred}(\mathbf{Z}) - \mathbf{Y}\|^2 \tag{3}$$

where Rec and Pred denote the reconstruction and prediction heads, respectively. Note that the decoder heads are solely used during training to shape the latent space and are discarded during inference.

4.3 Aggregating node- and scene-level representations

Let $\mathbf{z}_{s,i} \in \mathbb{R}^L$ denote the representation of agent i in scene s, where scene s contains n_s agents. The scene-level mean and covariance are computed as

$$\boldsymbol{\mu}_s = \frac{1}{n_s}\sum_{i=1}^{n_s}\mathbf{z}_{s,i}, \qquad \boldsymbol{\Sigma}_s = \frac{1}{n_s - 1}\sum_{i=1}^{n_s}\left(\mathbf{z}_{s,i} - \boldsymbol{\mu}_s\right)\left(\mathbf{z}_{s,i} - \boldsymbol{\mu}_s\right)^{\mathsf{T}}. \tag{4}$$

The aggregated dataset-level mean and covariance for a dataset $\mathcal{D}$ consisting of S scenes are defined as

$$\bar{\boldsymbol{\mu}}_{\mathcal{D}} = \frac{1}{S}\sum_{s=1}^{S}\boldsymbol{\mu}_s, \qquad \boldsymbol{\Sigma}_{\mathcal{D}} = \underbrace{\frac{1}{S}\sum_{s=1}^{S}\left(\boldsymbol{\mu}_s - \bar{\boldsymbol{\mu}}_{\mathcal{D}}\right)\left(\boldsymbol{\mu}_s - \bar{\boldsymbol{\mu}}_{\mathcal{D}}\right)^{\mathsf{T}}}_{\text{Between-scene cov.}} + \underbrace{\frac{1}{S}\sum_{s=1}^{S}\boldsymbol{\Sigma}_s}_{\text{Within-scene cov.}}. \tag{5}$$

The first covariance term captures variation between scene-level means, whereas the second captures the average variation among agents within each scene. Together, these terms provide a characterization of the dataset's latent structure and enable direct comparisons between datasets using probabilistic measures.

However, the empirical covariance may be ill-conditioned or singular, particularly in high-dimensional latent spaces with limited data. To improve numerical stability, a low-rank approximation with additive jitter regularization is used. Let $\boldsymbol{\Sigma}_{\mathcal{D}} = \mathbf{U}\Lambda\mathbf{U}^{\top}$ denote the eigendecomposition of the covariance matrix, where the columns of $\mathbf{U}$ are the eigenvectors and Λ is a diagonal matrix containing the corresponding eigenvalues in descending order. The regularized covariance is defined as

$$\widetilde{\boldsymbol{\Sigma}}_{\mathcal{D}} = \mathbf{U}_r \operatorname{diag}\left(\lambda_1, \dots, \lambda_r\right)\mathbf{U}_r^{\top} + \varepsilon\mathbf{I}, \qquad \varepsilon = \alpha\frac{\operatorname{tr}(\boldsymbol{\Sigma}_{\mathcal{D}})}{L}, \tag{6}$$

where $\mathbf{U}_r$ contains the eigenvectors associated with the r largest eigenvalues. The jitter parameter ε is scaled according to the average marginal variance, where α is a hyperparameter set to 3×10^{-3} and r is set to 16, both values chosen based on empirical results (see the supplementary material for more details).

5 Evaluation and results

This section evaluates the proposed framework. The datasets are introduced in Sec. 5.1, followed by implementation details in Sec. 5.2. The embedding model is assessed in Sec. 5.3 through complementary quantitative and qualitative latent space analyses. Section 5.4 investigates how well latent distances predict cross-dataset generalization under zero-shot and fine-tuning settings. Finally, an ablation on the latent dimension L analyzes its influence on transferability prediction accuracy.

5.1 Datasets

The investigations are conducted on a diverse set of trajectory-prediction datasets spanning pedestrian, vehicle, and mixed-traffic domains, covering both map-based and map-free settings, and collected using static cameras, drones, or instrumented vehicles. An overview is provided in Tab. 1, where datasets are grouped by collection method and listed chronologically by release year.

All datasets are converted to a unified format using the `Dronalize` toolbox [93], including resampling to 10 Hz and adopting a common feature representation. Official train/validation/test splits are used when available; otherwise, the splitting protocol in `Dronalize` is applied. For cross-dataset evaluation, all models are evaluated using a common 3 s prediction horizon, determined by the shortest fixed horizon among datasets with official test splits (*e.g.*, View-of-Delft). During training, however, models predict up to a maximum horizon of 8 s, with supervision provided where future trajectories are available.

Following common practice [2, 39], the ETH and UCY datasets are organized into the five standard evaluation subsets: *eth*, *hotel*, *univ*, *zara1*, and *zara2*. Each subset is treated as a separate dataset throughout the experiments. Accordingly, the evaluation comprises 24 datasets in total.

Table 1: Overview of the 24 trajectory-prediction datasets used in the experiments. Locations are condensed to country. Map info indicates whether HD maps (✓), no maps (✗), or partial map data are provided. The datasets balance prevalence in the literature with complementary characteristics, including geographic coverage and agent composition.

Name	Year	Location	Agents	Sensor	Map Info	Size
I-80 [85]	2006	USA	Vehicles	Camera	Lane lines	23 480
US-101 [86]	2007	USA	Vehicles	Camera	Lane lines	18 683
univ [55]	2007	Cyprus	Pedestrians	Camera	✗	4196
zara1 [55]	2007	Cyprus	Pedestrians	Camera	✗	4096
zara2 [55]	2007	Cyprus	Pedestrians	Camera	✗	4103
eth [76]	2009	Switzerland	Pedestrians	Camera	✗	4013
hotel [76]	2009	Switzerland	Pedestrians	Camera	✗	4040
ApolloScape [65]	2019	China	Mixed	Vehicle	✗	49 361
Argoverse [13]	2019	USA	Mixed	Vehicle	✓	323 557
nuScenes [12]	2020	USA, Singapore	Mixed	Vehicle	✓	195 103
Lyft Level 5 [42]	2021	USA	Mixed	Vehicle	✓	292 329
WOMD [22]	2021	USA	Mixed	Vehicle	✓	576 012
Argoverse 2 [96]	2023	USA	Mixed	Vehicle	✓	249 880
View-of-Delft [9]	2024	Netherlands	Mixed	Vehicle	✓	10 486
highD [52]	2018	Germany	Vehicles	Drone	Lane lines	129 786
INTERACTION [103]	2019	USA, China, Germany, Bulgaria	Mixed	Drone	✓	62 022
inD [8]	2020	Germany	Mixed	Drone	✓	155 609
rounD [53]	2020	Germany	Mixed	Drone	✓	111 973
openDD [11]	2020	Germany	Mixed	Drone	✓	370 073
exiD [71]	2022	Germany	Vehicles	Drone	✓	311 309
SIND [98]	2022	China	Mixed	Drone	✓	260 484
AD4CHE [105]	2023	China	Vehicles	Drone	Images	41 333
uniD [56]	2024	Germany	Mixed	Drone	✓	189 023
A43 [6]	2024	Germany	Vehicles	Drone	Lane lines	44 565

5.2 Implementation details

The embedding model was implemented in PyTorch [75] and PyTorch Geometric [28]. It was trained on a single NVIDIA A100 GPU using the AdamW optimizer [64] with a batch size of 512. To mitigate the imbalance in dataset sizes when jointly training on all datasets, mini-batches were constructed using weighted sampling, with each sample $j \in \mathcal{D}_i$ assigned the weight

$$w_j \propto \frac{1}{|\mathcal{D}_i|^\alpha}, \tag{7}$$

where $\mathcal{D}_i$ denotes the i-th dataset. Here, $\alpha = 0$ corresponds to sampling proportional to dataset size and $\alpha = 1$ to uniform sampling across datasets. A value of $\alpha = 0.5$ is used, reducing the dominance of larger datasets while preserving a moderate bias toward them. Training was conducted for 100 epochs, with an initial learning rate of 10^{-3}, which was decayed to 10^{-5} using a cosine annealing schedule [63]. Teacher forcing was used for both heads during the first 25 epochs, with the probability annealed linearly from 1 to 0. The final models used in all the experiments were selected based on minimum ADE performance on the validation set.

5.3 Embedding evaluation

In the latent space, each dataset is modeled as a multivariate Gaussian distribution defined by its dataset-level mean and covariance. Let $\mathcal{I} = \{1, \dots, K\}$ index the datasets $\mathcal{D}_1, \dots, \mathcal{D}_K$. For $i, j \in \mathcal{I}$, the directed dissimilarity of $\mathcal{D}_i$ from $\mathcal{D}_j$ is defined as the KL divergence between their corresponding Gaussian distributions:

$$D_{\mathrm{KL}}(\mathcal{D}_i \| \mathcal{D}_j) = D_{\mathrm{KL}}\Big(\mathcal{N}\Big(\bar{\boldsymbol{\mu}}_{\mathcal{D}_i}, \widetilde{\boldsymbol{\Sigma}}_{\mathcal{D}_i}\Big) \,\Big\|\, \mathcal{N}\Big(\bar{\boldsymbol{\mu}}_{\mathcal{D}_j}, \widetilde{\boldsymbol{\Sigma}}_{\mathcal{D}_j}\Big)\Big), \qquad i, j \in \mathcal{I}, \tag{8}$$

where $D_{\mathrm{KL}}(\mathcal{D}_i \| \mathcal{D}_j)$ measures how well the distribution of dataset $\mathcal{D}_j$ approximates that of dataset $\mathcal{D}_i$. Since the KL divergence is asymmetric, the resulting dissimilarity captures directional differences in dataset coverage and variability.

Figure 3 presents a heatmap of the KL divergence between embedding distributions for each dataset pair, where lower values indicate greater latent similarity. Several interesting patterns emerge. For example, nuScenes appears *easier* to approximate using other instrumented-vehicle datasets such as Lyft L5, WOMD, Argoverse, and Argoverse 2 than vice versa. For instance,

$$D_{\mathrm{KL}}(\text{nuScenes}\|\text{WOMD}) = 358, \quad D_{\mathrm{KL}}(\text{WOMD}\|\text{nuScenes}) = 36602, \tag{9}$$

indicating that these datasets contain information that transfers to nuScenes more effectively than nuScenes transfers to them. This asymmetry aligns with the empirical findings in both [25] and [33], where models pretrained on WOMD, Argoverse, or Argoverse 2 and evaluated on nuScenes achieve notably stronger performance than the reverse. The t-SNE [66] projection of scene latents in Fig. 4a supports this: nuScenes occupies a distinct subregion within the other datasets'

	inD	rounD	openDD	SIND	uniD	INTERACTION	Argoverse	Argoverse 2	nuScenes	Lyft	ApolloScape	WOMD	View-of-Delft	highD	exiD	A43	AD4CHE	I-80	US-101	ETH	hotel	univ	zara1	zara2
inD		268	152	66.9	303	249	366	302	44.5	170	91.6	285	378	6.86k	2.81k	12.4k	313	820	840	12.0k	6.38k	6.62k	5.91k	5.83k
rounD	1.16k		128	1.03k	1.87k	568	864	602	1.30k	473	682	1.30k	2.14k	16.5k	6.13k	12.0k	1.63k	2.02k	1.70k	25.3k	18.5k	19.0k	18.1k	17.6k
openDD	698	118		632	1.13k	540	785	630	763	407	417	1.21k	1.53k	13.5k	5.86k	11.0k	1.23k	1.58k	1.37k	22.2k	15.8k	16.2k	15.3k	15.1k
SIND	57.3	149	87.6		201	190	363	292	65.7	151	91.9	296	408	7.36k	3.35k	11.3k	533	951	903	9.01k	5.97k	6.07k	5.80k	5.67k
uniD	41.6	568	131	106		310	843	608	49.8	411	74.1	655	316	9.48k	4.27k	12.3k	442	734	825	2.84k	1.91k	1.91k	1.83k	1.78k
INTERACTION	9.89k	2.50k	3.50k	9.09k	13.6k		395	592	8.35k	2.73k	4.89k	276	13.5k	13.5k	6.50k	10.5k	3.62k	5.13k	4.60k	57.7k	41.7k	40.3k	41.9k	41.4k
Argoverse	28.7k	8.00k	10.5k	26.4k	38.5k	870		1.43k	23.6k	6.20k	13.0k	588	35.8k	18.5k	12.6k	15.8k	8.43k	12.8k	11.9k	136k	102k	97.4k	104k	101k
Argoverse 2	7.75k	1.87k	2.82k	7.20k	11.1k	624	406		6.40k	1.71k	3.57k	1.39k	10.1k	13.3k	6.73k	13.5k	2.62k	4.05k	3.86k	48.0k	36.0k	34.7k	36.0k	35.0k
nuScenes	69.5	236	129	72.0	201	190	315	275		134	50.6	358	263	6.32k	2.87k	11.7k	297	643	660	10.5k	6.10k	6.38k	5.70k	5.61k
Lyft	731	887	754	1.02k	2.20k	392	216	647	549		306	535	1.54k	6.72k	3.28k	11.5k	555	1.41k	1.47k	61.2k	29.2k	31.2k	26.6k	26.6k
ApolloScape	246	462	305	241	630	394	432	654	153	297		970	653	7.73k	4.37k	9.25k	519	975	965	25.4k	14.0k	14.7k	13.1k	12.8k
WOMD	43.9k	9.52k	15.0k	40.8k	59.5k	670	367	1.96k	36.6k	8.91k	20.4k		55.6k	34.7k	18.3k	25.1k	13.3k	19.9k	18.2k	171k	140k	132k	144k	141k
View-of-Delft	172	204	154	150	175	209	249	380	133	173	72.5	330		5.39k	2.86k	10.9k	255	470	492	4.95k	2.87k	2.87k	2.75k	2.65k
highD	12.1k	13.3k	16.9k	16.5k	42.4k	10.7k	8.92k	9.73k	22.2k	7.31k	9.13k	4.53k	42.0k		155	65.7	3.95k	14.1k	18.9k	396k	337k	334k	359k	341k
exiD	8.39k	10.5k	12.9k	12.7k	31.5k	6.62k	7.67k	6.85k	14.2k	4.81k	6.26k	3.17k	31.6k	1.21k		874	2.40k	7.79k	10.6k	380k	256k	262k	264k	257k
A43	7.10k	9.77k	11.1k	11.5k	26.8k	5.64k	4.97k	5.89k	11.3k	3.56k	5.32k	2.33k	23.3k	402	474		1.74k	6.74k	9.16k	380k	231k	241k	233k	227k
AD4CHE	917	1.74k	1.31k	1.23k	2.42k	485	540	1.24k	769	352	379	473	2.23k	2.05k	1.93k	2.13k		71.6	68.7	116k	47.7k	54.1k	43.9k	44.4k
I-80	688	1.11k	805	834	1.59k	601	523	784	653	516	251	944	1.51k	2.44k	3.23k	1.28k	71.7		20.2	71.1k	35.8k	39.6k	33.4k	33.6k
US-101	1.23k	3.30k	2.27k	2.47k	4.20k	777	520	978	1.05k	293	606	870	3.43k	2.02k	2.04k	715	64.3	32.2		234k	94.4k	109k	85.9k	87.7k
ETH	64.6	389	134	114	42.6	253	547	409	47.2	296	81.6	458	161	7.06k	2.70k	12.6k	185	348	450		2.70	3.11	3.41	3.72
hotel	62.6	450	137	121	40.8	271	613	459	47.2	334	80.2	516	164	7.61k	2.89k	12.8k	195	348	463	12.3		0.68	0.62	0.70
univ	58.9	478	139	120	36.7	273	643	470	45.7	351	84.3	525	166	8.14k	3.06k	13.3k	203	386	507	14.6	0.61		1.38	1.60
zara1	60.8	356	123	101	40.9	238	505	383	45.2	272	78.0	426	158	6.68k	2.55k	12.5k	169	329	424	13.1	0.55	1.30		1.12
zara2	67.6	465	136	126	42.7	273	619	491	48.9	340	75.2	541	170	7.69k	2.93k	12.7k	200	326	449	16.3	0.61	1.48	1.05	

Fig. 3: KL divergence $D_{\mathrm{KL}}(\mathrm{row}\|\mathrm{col})$ between dataset embedding distributions. Entries indicate how well column datasets approximate row datasets. Lower values denote closer alignment, typically correlating with stronger zero-shot transfer from column training datasets to row evaluation datasets. Matrix asymmetry reflects directional differences in dataset coverage and variability.

broader clusters. This indicates that the asymmetric relationship observed in the high-dimensional latent space is also reflected in a low-dimensional projection.

Highway datasets are more similar to each other than to urban datasets, though asymmetrically. The t-SNE projection in Fig. 4b clearly shows overlap among highD, exiD, and A43, all collected in Germany, likely reflecting regional driving patterns. Furthermore, AD4CHE, I-80, and US-101 cluster together, suggesting the latent space encodes shared behavioral or traffic-flow characteristics beyond geographic proximity.

Another important observation is the large divergence between the pedestrian-focused datasets and the rest. The results indicate that the pedestrian datasets are substantially different from the others, suggesting low potential for direct transferability. Partial exceptions are uniD and View-of-Delft, which are heavily pedestrian-oriented and appear to be closer in the embedding space.

Most interestingly, the embedding space also uncovers relationships between datasets that do not share presumed commonalities, such as acquisition method or agent type. For instance, several drone-acquired datasets, including inD, openDD, SIND, INTERACTION, exhibit low KL divergence with instrumented-vehicle datasets such as nuScenes, Argoverse 2, and WOMD. This suggests that cross-domain transfer is possible even between heterogeneous datasets. This is significant, as trajectory-prediction research often suffers from scarce, domain-

(a) Urban datasets

(b) Highway datasets

Fig. 4: Two-dimensional t-SNE visualization of the learned scenario embeddings for (a) a subset of the urban datasets (acquired with instrumented vehicles) and (b) highway datasets, with contour lines indicating density.

specific data [87], indicating untapped potential in leveraging datasets that are comparatively underused in the motion prediction literature.

5.4 Transferability evaluation

To evaluate the interpretation of *closeness* in the latent space as a measure of transferability, a series of experiments is conducted in which a trajectory-prediction model is trained on one dataset and evaluated on another. Motivated by its state-of-the-art performance and its architectural **dissimilarity** to the proposed embedding model, QCNet [109] is adopted as the baseline predictor, following the original training objectives and inference procedure. To ensure comparability across datasets, a lightweight input transformation layer is introduced, and model capacity is reduced to maintain computational feasibility across all datasets (see the supplementary material for more details). Model performance is reported using the minimum average positional displacement error for the best of $K = 6$ predicted trajectories (minADE_6) over a 3-second horizon.

Zero-shot transferability To evaluate cross-dataset generalization, a model is trained on one dataset and is evaluated, without any adaptation, on all remaining datasets. With 24 datasets, this yields $24 \times 23 = 552$ transfer pairs.

Using the divergence introduced in (8), it is investigated how well this measure predicts zero-shot transfer performance. To this end, minADE_6 is examined against $D_{\text{KL}}(\mathcal{D}_e \| \mathcal{D}_t)$, where $\mathcal{D}_e$ and $\mathcal{D}_t$ denote evaluation and training datasets, respectively. As shown in Fig. 5, pairs with larger divergence exhibit consistently higher minADE_6, indicating poorer transfer.

A Spearman's rank correlation analysis confirms a strong positive association: increasing divergence aligns with reduced zero-shot performance. Overall, the KL divergence achieves a rank correlation of $\rho = 0.811$, with a 95% confidence interval of (0.782, 0.840).

Transferability under fine-tuning Beyond zero-shot performance, it is important to understand how datasets support transfer when used for fine-tuning

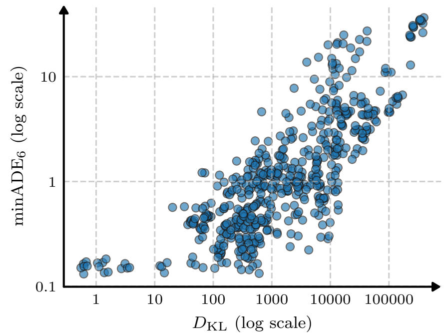


Fig. 5: KL divergence versus zero-shot minADE$_6$ transfer performance across all 552 dataset pairs. Larger divergence correlates with worse transferability.

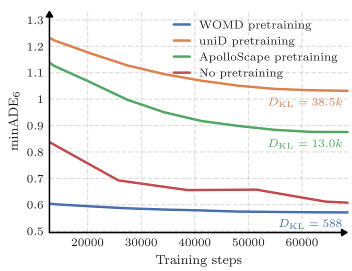


Fig. 6: Argoverse fine-tuning from different sources. Sources with lower divergence scores yield better adaptation.

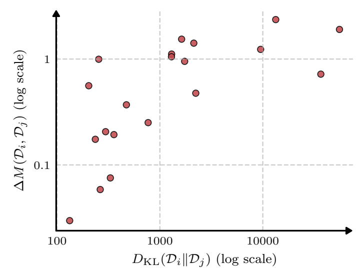


Fig. 7: KL divergence versus performance degradation after fine-tuning. Higher divergence generally corresponds to greater forgetting of the source dataset.

toward a target domain, as this has direct practical utility. This is studied from two complementary perspectives: whether the latent divergence can guide pretraining source selection, and whether it predicts how much source knowledge is lost during adaptation.

Pretraining source selection. As a practical case study, Argoverse is used as the target dataset. Models pretrained on a set of candidate source datasets are subsequently fine-tuned on Argoverse, and target performance is recorded after adaptation. As shown in Fig. 6, sources with lower KL divergence to Argoverse generally yield stronger post-fine-tuning performance on the target domain, indicating that the latent metric can serve as a practical guide for pretraining source selection, even before any task-specific training is conducted.

Catastrophic forgetting. A complementary concern when fine-tuning is catastrophic forgetting [35]: the tendency for adaptation to a new domain to overwrite previously acquired source knowledge. To examine this, models pretrained on

Table 2: Spearman's rank correlation coefficient ρ for different dataset dissimilarity measures and latent dimensions. **Bold** denotes the best measure for each task–dimension combination, where higher is better. For KL divergence, 95% confidence intervals are shown below the corresponding estimates.

	Zero-shot transfer				Post-fine-tuning degradation			
Measure	$L = 16$	$L = 32$	$L = 64$	$L = 128$	$L = 16$	$L = 32$	$L = 64$	$L = 128$
L1	0.475	0.468	0.470	0.425	0.136	0.254	0.226	0.145
Wasserstein	0.489	0.482	0.482	0.448	0.269	0.296	0.335	0.257
MMD	0.410	0.412	0.410	0.362	0.136	0.245	0.193	0.185
KL	**0.781** (.750, .810)	**0.811** (.782, .840)	**0.746** (.711, .783)	**0.736** (.681, .760)	**0.504** (.045, .792)	**0.729** (.403, .913)	**0.794** (.578, .922)	**0.874** (.639, .970)

a source dataset $\mathcal{D}_i$ are fine-tuned on a target dataset $\mathcal{D}_j$, and performance is subsequently evaluated on $\mathcal{D}_i$. Let $m_{\text{org}}^{\mathcal{D}_i}$ and $m_{\text{ft}}^{\mathcal{D}_j}$ denote the original and fine-tuned models, respectively. The performance change is defined as:

$$\Delta M(\mathcal{D}_i, \mathcal{D}_j) = M(m_{\text{ft}}^{\mathcal{D}_j}, \mathcal{D}_i) - M(m_{\text{org}}^{\mathcal{D}_i}, \mathcal{D}_i), \tag{10}$$

where $M(\cdot, \cdot)$ denotes minADE$_6$ evaluated on a specific dataset. Lower ΔM indicates less forgetting, and it is expected to correlate with $D_{\text{KL}}(\mathcal{D}_i \| \mathcal{D}_j)$.

Experiments across five datasets, AD4CHE, nuScenes, rounD, View-of-Delft, and WOMD, produce 20 transfer pairs. Figure 7 indicates a positive trend: larger divergence between datasets generally coincides with stronger forgetting. Spearman's rank correlation is $\rho = 0.729$, with a 95% confidence interval of (0.403, 0.913), confirming a strong positive association. Taken together, the two results show that the latent divergence is predictive of fine-tuning behavior from both directions: it anticipates adaptation quality before training, and captures the degree of source knowledge loss afterward.

Motivation for the Gaussian assumption To assess the Gaussian modeling assumption, Tab. 2 compares the predictive performance of the KL divergence with alternative similarity measures, including the non-parametric maximum mean discrepancy (MMD) [37] computed directly on scene-level embeddings. The KL divergence consistently achieves the highest rank correlation across all evaluated latent dimensionalities in both zero-shot and fine-tuning tasks, empirically justifying the Gaussian approximation.

Ablation on latent dimension size Finally, an ablation study examines how the latent dimensionality L influences the predictive quality of the learned embeddings. Models with $L \in \{16, 32, 64, 128\}$ are evaluated on both zero-shot transfer and post-fine-tuning degradation tasks. As shown in Tab. 2, the two tasks exhibit opposing trends. For zero-shot transfer, the correlation decreases from $\rho = 0.811$ at $L = 32$ to $\rho = 0.736$ at $L = 128$, indicating that the more compact representation better preserves the ordering of cross-dataset transfer

Table 3: Definitions of dataset characteristics and derived correction terms $\gamma_{i,j}^{(\ell)}$ for transfer from source $\mathcal{D}_i$ to target $\mathcal{D}_j$. Scalar corrections are computed as the ratio of source to target values, whereas the agent class distribution $\mathbf{p} \in \mathbb{R}^2$ use L_1 distance.

Term	Description	Symbol	Calculation of $\gamma_{i,j}^{(\ell)}$
Size	Total number of samples in the dataset	N	$N_i \,/\, N_j$
Speed	Mean agent speed across samples and time steps	$\bar{v}$	$\bar{v}_i \,/\, \bar{v}_j$
#agents	Mean number of agents present in each sample	$\bar{n}$	$\bar{n}_i \,/\, \bar{n}_j$
Types	Mean empirical distribution over agent classes	$\bar{\mathbf{p}}$	$\lVert \bar{\mathbf{p}}_i - \bar{\mathbf{p}}_j \rVert_1$
Time	Mean duration of the observed input sequence	$\bar{t}$	$\bar{t}_i \,/\, \bar{t}_j$

performance. In contrast, the correlation for post-fine-tuning degradation increases from $\rho = 0.729$ to $\rho = 0.874$, suggesting that larger latent spaces may better capture differences associated with adaptation and forgetting. However, the wide and overlapping confidence intervals, together with the smaller number of fine-tuning pairs, warrant a more cautious interpretation of this trend.

5.5 Investigation into latent space quality

To evaluate whether latent dataset embeddings capture complex discrepancies beyond basic statistical heuristics, transfer performance $T_{i,j}$ from source dataset $\mathcal{D}_i$ to target $\mathcal{D}_j$ is modeled using the KL divergence and a set of candidate explicit heuristic metrics $\gamma_{i,j}^{(\ell)}$

$$T_{i,j} \approx a\, D_{\mathrm{KL}}(\mathcal{D}_j \| \mathcal{D}_i) + \sum_{\ell \in \mathcal{S}} b_\ell\, \gamma_{i,j}^{(\ell)} + c. \tag{11}$$

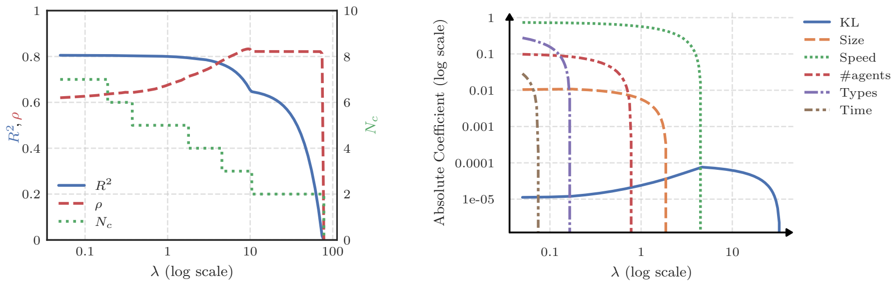
Here, $\mathcal{S}$ encompasses five dataset differences, with definitions and correction ratios detailed in Tab. 3. To isolate the most informative predictors, this model is optimized using Lasso regularization, applying an L_1 penalty to the coefficients a and b_ℓ. Formally, the parameters $\theta = \{a, \{b_\ell\}_{\ell \in \mathcal{S}}, c\}$ are estimated by minimizing

$$\min_\theta \sum_{i,j} \Big(T_{i,j} - a\, D_{\mathrm{KL}}(\mathcal{D}_j \| \mathcal{D}_i) - \sum_{\ell \in \mathcal{S}} b_\ell\, \gamma_{i,j}^{(\ell)} - c \Big)^2 + \lambda \Big(|a| + \sum_{\ell \in \mathcal{S}} |b_\ell| \Big). \tag{12}$$

This formulation enforces sparsity. Coefficients driven to zero correspond to predictors that do not contribute substantially to explaining transfer performance.

The regression results in Fig. 8 demonstrate the representational capacity of the learned latent space. In Fig. 8b, the KL divergence coefficient remains nonzero for much higher values of the regularization strength λ compared to the other coefficients, indicating that the learned representation captures these underlying patterns.

The exception is average agent speed, as its coefficient remains nonzero under stronger regularization. Nevertheless, using the speed factor by itself achieves a lower rank correlation, $\rho = 0.624$, with a 95% confidence interval of (0.561, 0.669),



(a) **(b)**

Fig. 8: (a) Evolution of coefficient of determination R^2, Spearman's ρ, and the number of selected correction terms N_c against the regularization parameter λ. (b) Coefficient paths for all terms in (11).

compared with $\rho = 0.811$ for the KL divergence, with a 95% confidence interval of (0.782, 0.840). This further indicates that the latent representation captures discrepancies that are not adequately described by the considered statistical heuristics.

6 Conclusion

A latent embedding framework that systematically characterizes trajectory datasets and quantifies their transferability was introduced. The framework reveals structured relationships between datasets and offers guidance on selecting effective pretraining sources. Empirical results across 24 diverse datasets show that distances in the latent space strongly correlate with both zero-shot generalization and fine-tuning outcomes, demonstrating that transferability can be predicted rather than discovered through exhaustive experimentation. The analysis further identifies datasets that are less commonly used in prior work but still hold potential to improve performance on widely used motion-prediction benchmarks. These findings offer practical guidance for dataset selection and model development, contributing toward more robust and generalizable motion prediction systems.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable suggestions. This research was supported by the Strategic Research Area at Linköping-Lund in Information Technology (ELLIIT) and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Computations were enabled by the Berzelius resource provided by the Knut and Alice Wallenberg Foundation at the National Supercomputer Centre.

References

1. Achille, A., Lam, M., Tewari, R., Ravichandran, A., Maji, S., Fowlkes, C.C., Soatto, S., Perona, P.: Task2Vec: Task embedding for meta-learning. In: IEEE/CVF Int. Conf. Comput. Vision. pp. 6430–6439 (2019)
2. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social LSTM: Human trajectory prediction in crowded spaces. In: IEEE Conf. Comput. Vision Pattern Recog. pp. 961–971 (2016)
3. Amirian, J., Hayet, J.B., Pettré, J.: Social ways: Learning multi-modal distributions of pedestrian trajectories with GANs. In: IEEE/CVF Conf. Comput. Vision Pattern Recog. (2019)
4. Amirian, J., Zhang, B., Castro, F.V., Baldelomar, J.J., Hayet, J.B., Pettré, J.: OpenTraj: Assessing prediction complexity in human trajectories datasets. In: Asian Conf. Comput. Vision (2020)
5. Bae, I., Park, Y.J., Jeon, H.G.: SingularTrajectory: Universal trajectory predictor using diffusion model. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024)
6. Berghaus, M., Lamberty, S., Ehlers, J., Kalló, E., Oeser, M.: Vehicle trajectory dataset from drone videos including off-ramp and congested traffic–analysis of data quality, traffic flow, and accident risk. Communic. in Transport. Research **4**, 100133 (2024)
7. Bishop, C.M., Bishop, H.: Deep learning. Springer (2024)
8. Bock, J., Krajewski, R., Moers, T., Runde, S., Vater, L., Eckstein, L.: The inD dataset: A drone dataset of naturalistic road user trajectories at German intersections. In: IEEE Intell. Veh. Symp. pp. 1929–1934 (2020)
9. Boekema, H.J.H., Martens, B.K., Kooij, J.F., Gavrila, D.M.: Multi-class trajectory prediction in urban traffic using the view-of-delft prediction dataset. IEEE Robot. Automat. Lett. **9**(5), 4806–4813 (2024)
10. Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258 (2021)
11. Breuer, A., Termöhlen, J.A., Homoceanu, S., Fingscheidt, T.: openDD: A large-scale roundabout drone dataset. In: IEEE Intell. Transport. Syst. Conf. pp. 1–6. IEEE (2020)
12. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuScenes: A multimodal dataset for autonomous driving. In: IEEE/CVF Conf. Comput. Vision Pattern Recog. pp. 11621–11631 (2020)
13. Chang, M.F., Lambert, J., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., Wang, D., Carr, P., Lucey, S., Ramanan, D., et al.: Argoverse: 3D tracking and forecasting with rich maps. In: IEEE/CVF Conf. Comput. Vision Pattern Recog. pp. 8748–8757 (2019)
14. Cho, K., van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder–decoder approaches. In: SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation. pp. 103–111 (2014)
15. Choi, Y., Mercurius, R.C., Shabestary, S.M.A., Rasouli, A.: DICE: Diverse diffusion model with scoring for trajectory prediction. In: IEEE Intell. Veh. Symp. pp. 3023–3029 (2024)

16. Clark, R.: Predicting Transfer Learning Performance Using Dataset Similarity for Time Series Classification of Human Activity Recognition. Ph.D. thesis, McMaster University (2022)
17. Cui, H., Radosavljevic, V., Chou, F.C., Lin, T.H., Nguyen, T., Huang, T.K., Schneider, J., Djuric, N.: Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In: IEEE Int. Conf. Robot. Automat. pp. 2090–2096 (2019)
18. Deo, N., Trivedi, M.M.: Convolutional social pooling for vehicle trajectory prediction. In: IEEE Conf. Comput. Vision Pattern Recog. pp. 1468–1476 (2018)
19. Deo, N., Wolff, E., Beijbom, O.: Multimodal trajectory prediction conditioned on lane-graph traversals. In: Conf. on Robot Learn. pp. 203–212. PMLR (2022)
20. Diehl, F., Brunner, T., Le, M.T., Knoll, A.: Graph neural networks for modelling traffic participant interaction. In: IEEE Intell. Veh. Symp. pp. 695–701 (2019)
21. Ehrig, C., Sonnleitner, B., Neumann, U., Cleophas, C., Forestier, G.: The impact of data set similarity and diversity on transfer learning success in time series forecasting. arXiv preprint arXiv:2404.06198 (2024)
22. Ettinger, S., Cheng, S., Caine, B., Liu, C., Zhao, H., Pradhan, S., Chai, Y., Sapp, B., Qi, C.R., Zhou, Y., et al.: Large scale interactive motion forecasting for autonomous driving: The Waymo open motion dataset. In: IEEE/CVF Int. Conf. Comput. Vision. pp. 9710–9719 (2021)
23. Fang, J., Wang, F., Xue, J., Chua, T.S.: Behavioral intention prediction in driving scenes: A survey. IEEE Transactions on Intelligent Transportation Systems **25**(8), 8334–8355 (2024)
24. Feng, K., Li, C., Ren, D., Yuan, Y., Wang, G.: On the road to portability: Compressing end-to-end motion planner for autonomous driving. In: IEEE/CVF Conf. Comput. Vision Pattern Recog. pp. 15099–15108 (2024)
25. Feng, L., Bahari, M., Amor, K.M.B., Zablocki, É., Cord, M., Alahi, A.: UniTraj: A unified framework for scalable vehicle trajectory prediction. In: Eur. Conf. Comput. Vision. pp. 106–123. Springer (2024)
26. Feng, L., Li, Q., Peng, Z., Tan, S., Zhou, B.: TrafficGen: Learning to generate diverse and realistic traffic scenarios. In: IEEE Int. Conf. Robot. Automat. pp. 3567–3575. IEEE (2023)
27. Feng, L., Tung, F., Ahmed, M.O., Bengio, Y., Hajimirsadegh, H.: Were RNNs All We Needed? arXiv preprint arXiv:2410.01201 (2024)
28. Fey, M., Lenssen, J.E.: Fast graph representation learning with PyTorch Geometric. In: ICLR Workshop on Representation Learn. on Graphs and Manifolds (2019)
29. Gao, H., Wang, Z., Li, Y., Long, K., Yang, M., Shen, Y.: A survey for foundation models in autonomous driving. arXiv preprint arXiv:2402.01105 (2024)
30. Gao, X., Jia, X., Li, Y., Xiong, H.: Dynamic scenario representation learning for motion forecasting with heterogeneous graph convolutional recurrent networks. IEEE Robot. Automat. Lett. **8**(5), 2946–2953 (2023)
31. Geng, M., Li, J., Li, C., Xie, N., Chen, X., Lee, D.H.: Adaptive and simultaneous trajectory prediction for heterogeneous agents via transferable hierarchical transformer network. IEEE Trans. Intell. Transport. Syst. **24**(10), 11479–11492 (2023)
32. Gilles, T., Sabatini, S., Tsishkou, D., Stanciulescu, B., Moutarde, F.: GOHOME: Graph-oriented heatmap output for future motion estimation. In: IEEE Int. Conf. Robot. Automat. pp. 9107–9114. IEEE (2022)
33. Gilles, T., Sabatini, S., Tsishkou, D., Stanciulescu, B., Moutarde, F.: Uncertainty estimation for cross-dataset performance in trajectory prediction. In: Workshop on

Fresh Perspectives on the Future of Autonomous Driving, IEEE Int. Conf. Robot. Automat. (2022)
34. Giuliari, F., Hasan, I., Cristani, M., Galasso, F.: Transformer networks for trajectory forecasting. In: 25th international conference on pattern recognition (ICPR). pp. 10335–10342. IEEE (2021)
35. Goodfellow, I.J., Mirza, M., Xiao, D., Courville, A., Bengio, Y.: An empirical investigation of catastrophic forgetting in gradient-based neural networks. arXiv preprint arXiv:1312.6211 (2013)
36. Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: A survey. International journal of computer vision **129**(6), 1789–1819 (2021)
37. Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. The Journal of Machine Learning Research **13**(1), 723–773 (2012)
38. Gu, T., Chen, G., Li, J., Lin, C., Rao, Y., Zhou, J., Lu, J.: Stochastic trajectory prediction via motion indeterminacy diffusion. In: IEEE/CVF Conf. Comput. Vision Pattern Recog. pp. 17113–17122. IEEE/CVF (2022)
39. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social GAN: Socially acceptable trajectories with generative adversarial networks. In: IEEE Conf. Comput. Vision Pattern Recog. (2018)
40. Hegde, D., Yasarla, R., Cai, H., Han, S., Bhattacharyya, A., Mahajan, S., Liu, L., Garrepalli, R., Patel, V.M., Porikli, F.: Distilling multi-modal large language models for autonomous driving. In: IEEE/CVF Conf. Comput. Vision Pattern Recog. pp. 27575–27585 (2025)
41. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
42. Houston, J., Zuidhof, G., Bergamini, L., Ye, Y., Chen, L., Jain, A., Omari, S., Iglovikov, V., Ondruska, P.: One thousand and one hours: Self-driving motion prediction dataset. In: Conference on Robot Learning (CoRL). pp. 409–418. PMLR (2021)
43. Hu, Y., Zhan, W., Tomizuka, M.: Probabilistic prediction of vehicle semantic intention and motion. In: IEEE Intell. Veh. Symp. pp. 307–313 (2018)
44. Hu, Y., Zhan, W., Tomizuka, M.: Scenario-transferable semantic graph reasoning for interaction-aware probabilistic prediction. IEEE Trans. Intell. Transport. Syst. **23**(12), 23212–23230 (2022)
45. Huang, Y., Du, J., Yang, Z., Zhou, Z., Zhang, L., Chen, H.: A survey on trajectory-prediction methods for autonomous driving. IEEE Trans. Intell. Veh. **7**(3), 652–674 (2022)
46. Huang, Z., Mo, X., Lv, C.: Multi-modal motion prediction with transformer-based neural network for autonomous driving. In: IEEE Int. Conf. Robot. Automat. pp. 2605–2611 (2022)
47. Ivanovic, B., Harrison, J., Pavone, M.: Expanding the deployment envelope of behavior prediction via adaptive meta-learning. In: IEEE Int. Conf. Robot. Automat. pp. 7786–7793. IEEE (2023)
48. Ivanovic, B., Song, G., Gilitschenski, I., Pavone, M.: trajdata: A unified interface to multiple human trajectory datasets. In: Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2023)
49. Jaipuria, N., Habibi, G., How, J.P.: Learning in the curbside coordinate frame for a transferable pedestrian trajectory prediction model. In: IEEE Intell. Transport. Syst. Conf. pp. 3125–3131 (2018)
50. Jiang, C., Cornman, A., Park, C., Sapp, B., Zhou, Y., Anguelov, D., et al.: MotionDiffuser: Controllable multi-agent motion prediction using diffusion. In: IEEE/CVF Conf. Comput. Vision Pattern Recog. pp. 9644–9653 (2023)

51. Kong, H., Xu, J., Gong, S., Yang, J., Zhang, S.: Adaptive pedestrian trajectory prediction via target-directed angle augmentation. In: IEEE Int. Conf. Acoust., Speech, Signal Processing. pp. 4065–4069. IEEE (2024)
52. Krajewski, R., Bock, J., Kloeker, L., Eckstein, L.: The highD dataset: A drone dataset of naturalistic vehicle trajectories on German highways for validation of highly automated driving systems. In: IEEE Intell. Transport. Syst. Conf. pp. 2118–2125 (2018)
53. Krajewski, R., Moers, T., Bock, J., Vater, L., Eckstein, L.: The rounD dataset: A drone dataset of road user trajectories at roundabouts in Germany. In: IEEE Intell. Transport. Syst. Conf. pp. 1–6 (2020)
54. Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., Carlini, N.: Deduplicating training data makes language models better. arXiv preprint arXiv:2107.06499 (2021)
55. Lerner, A., Chrysanthou, Y., Lischinski, D.: Crowds by example. In: Computer graphics forum. vol. 26, pp. 655–664. Wiley Online Library (2007)
56. leveLXData: The uniD Dataset: A university drone dataset. https://levelxdata.com/unid-dataset/ (2024), accessed: September 8, 2024
57. Li, J., Ma, H., Zhang, Z., Li, J., Tomizuka, M.: Spatio-temporal graph dual-attention network for multi-agent prediction and tracking. IEEE Trans. Intell. Transport. Syst. **23**(8), 10556–10569 (2022)
58. Li, Q., Peng, Z.M., Feng, L., Liu, Z., Duan, C., Mo, W., Zhou, B.: ScenarioNet: Open-source platform for large-scale traffic scenario simulation and modeling. Advances in Neural Information Processing Systems (NeurIPS) **36**, 3894–3920 (2023)
59. Li, X., Ying, X., Chuah, M.C.: GRIP++: Enhanced graph-based interaction-aware trajectory prediction for autonomous driving. arXiv preprint arXiv:1907.07792 (2019)
60. Liang, M., Yang, B., Hu, R., Chen, Y., Liao, R., Feng, S., Urtasun, R.: Learning lane graph representations for motion forecasting. In: Eur. Conf. Comput. Vision. pp. 541–556 (2020)
61. Liu, M., Cheng, H., Yang, M.Y.: Tracing the influence of predecessors on trajectory prediction. In: Int. Conf. on Comput. Vision Worksh. (ICCVW). pp. 3245–3255. IEEE/CVF (2023)
62. Liu, Y., Zhang, J., Fang, L., Jiang, Q., Zhou, B.: Multimodal motion prediction with stacked transformers. In: IEEE/CVF Conf. Comput. Vision Pattern Recog. pp. 7577–7586 (2021)
63. Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with warm restarts. In: International Conference on Learning Representations (ICLR) (2017)
64. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (ICLR) (2019)
65. Ma, Y., Zhu, X., Zhang, S., Yang, R., Wang, W., Manocha, D.: Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 6120–6127 (2019)
66. Maaten, L.v.d., Hinton, G.: Visualizing data using t-SNE. Journal of Machine Learning Research **9**, 2579–2605 (2008)
67. Mao, W., Xu, C., Zhu, Q., Chen, S., Wang, Y.: Leapfrog diffusion model for stochastic trajectory prediction. In: IEEE/CVF Conf. Comput. Vision Pattern Recog. pp. 5517–5526 (2023)
68. Marchetti, F., Becattini, F., Seidenari, L., Bimbo, A.D.: MANTRA: Memory augmented networks for multiple trajectory prediction. In: IEEE/CVF Conf. Comput. Vision Pattern Recog. pp. 7143–7152 (2020)

69. Messaoud, K., Deo, N., Trivedi, M.M., Nashashibi, F.: Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation. In: IEEE Intell. Veh. Symp. pp. 165–170 (2021)
70. Messaoud, K., Yahiaoui, I., Verroust-Blondet, A., Nashashibi, F.: Attention based vehicle trajectory prediction. IEEE Trans. Intell. Veh. **6**(1), 175–185 (2021)
71. Moers, T., Vater, L., Krajewski, R., Bock, J., Zlocki, A., Eckstein, L.: The exiD dataset: A real-world trajectory dataset of highly interactive highway scenarios in Germany. In: IEEE Intell. Veh. Symp. pp. 958–964 (2022)
72. Morris, C., Ritzert, M., Fey, M., Hamilton, W.L., Lenssen, J.E., Rattan, G., Grohe, M.: Weisfeiler and leman go neural: Higher-order graph neural networks. In: AAAI Conference on Artificial Intelligence. pp. 4602–4609 (2019)
73. Park, D., Jeong, J., Yoon, K.J.: Improving transferability for cross-domain trajectory prediction via neural stochastic differential equation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 10145–10154 (2024)
74. Park, S.H., Lee, G., Seo, J., Bhat, M., Kang, M., Francis, J., Jadhav, A., Liang, P.P., Morency, L.P.: Diverse and admissible trajectory forecasting through multimodal context understanding. In: Eur. Conf. Comput. Vision. pp. 282–298 (2020)
75. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: PyTorch: An imperative style, high-performance deep learning library. In: Int. Conf. Adv. in Neural Inf. Process. Syst. (2019)
76. Pellegrini, S., Ess, A., Schindler, K., Van Gool, L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In: International Conference on Computer Vision (ICCV). pp. 261–268. IEEE (2009)
77. Penedo, G., Kydlíček, H., Lozhkov, A., Mitchell, M., Raffel, C.A., Von Werra, L., Wolf, T., et al.: The fineweb datasets: Decanting the web for the finest text data at scale. Advances in Neural Information Processing Systems **37**, 30811–30849 (2024)
78. Rudenko, A., Palmieri, L., Huang, W., Lilienthal, A.J., Arras, K.O.: The atlas benchmark: An automated evaluation framework for human motion prediction. In: Int. Conf. on Robot. and Hum. Interact. Comm. (RO-MAN). pp. 636–643. IEEE (2022)
79. Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofighi, H., Savarese, S.: SoPhie: An attentive GAN for predicting paths compliant to social and physical constraints. In: IEEE/CVF Conf. Comput. Vision Pattern Recog. pp. 1349–1358 (2019)
80. Salzmann, T., Ivanovic, B., Chakravarty, P., Pavone, M.: Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In: Eur. Conf. Comput. Vision. pp. 683–700 (2020)
81. Shi, X., Zhang, H., Yuan, W., Shibasaki, R.: MetaTraj: Meta-learning for cross-scene cross-object trajectory prediction. IEEE Trans. Intell. Transport. Syst. **24**(12), 14000–14009 (2023)
82. Tolstaya, E., Mahjourian, R., Downey, C., Vadarajan, B., Sapp, B., Anguelov, D.: Identifying driver interactions via conditional behavior prediction. In: IEEE Int. Conf. Robot. Automat. pp. 3473–3479 (2021)
83. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
84. Ullrich, L., McMaster, A., Graichen, K.: Transfer learning study of motion transformer-based trajectory predictions. In: IEEE Intell. Veh. Symp. pp. 110–117 (2024)

85. U.S. Department of Transportation Federal Highway Administration: Next generation simulation (ngsim) vehicle trajectories and supporting data. i-80. http://doi.org/10.21949/1504477 (2016), dataset provided by ITS DataHub through Data.transportation.gov. Accessed 2025-06-09
86. U.S. Department of Transportation Federal Highway Administration: Next generation simulation (ngsim) vehicle trajectories and supporting data. us-101. http://doi.org/10.21949/1504477 (2016), dataset provided by ITS DataHub through Data.transportation.gov. Accessed 2025-06-09
87. Wang, L., Lavoie, M.A., Papais, S., Nisar, B., Chen, Y., Ding, W., Ivanovic, B., Shao, H., Abuduweili, A., Cook, E., et al.: Deployable and generalizable motion prediction: Taxonomy, open challenges and future directions. arXiv preprint arXiv:2505.09074 (2025)
88. Wang, Z., Guo, J., Zhang, H., Wan, R., Zhang, J., Pu, J.: Bridging the gap: Improving domain generalization in trajectory prediction. IEEE Trans. Intell. Veh. **9**(1), 1780–1791 (2023)
89. Wang, Z., Zhang, J., Chen, J., Zhang, H.: Spatio-temporal context graph transformer design for map-free multi-agent trajectory prediction. IEEE Trans. Intell. Veh. **9**(1), 1369–1381 (2024)
90. Wang, Z., Dai, Z., Póczos, B., Carbonell, J.: Characterizing and avoiding negative transfer. In: IEEE/CVF Conf. Comput. Vision Pattern Recog. pp. 11293–11302 (2019)
91. Westny, T., Frisk, E., Olofsson, B.: Vehicle behavior prediction and generalization using imbalanced learning techniques. In: IEEE Intell. Transport. Syst. Conf. pp. 2003–2010 (2021)
92. Westny, T., Olofsson, B., Frisk, E.: Diffusion-based environment-aware trajectory prediction. arXiv preprint arXiv:2403.11643 (2024)
93. Westny, T., Olofsson, B., Frisk, E.: Toward unified practices in trajectory prediction research on bird's eye view datasets. In: IEEE Intell. Veh. Symp. (2025)
94. Westny, T., Oskarsson, J., Olofsson, B., Frisk, E.: Evaluation of differentially constrained motion models for graph-based trajectory prediction. In: IEEE Intell. Veh. Symp. (2023)
95. Westny, T., Oskarsson, J., Olofsson, B., Frisk, E.: MTP-GO: Graph-based probabilistic multi-agent trajectory prediction with neural ODEs. IEEE Trans. Intell. Veh. **8**(9), 4223–4236 (2023)
96. Wilson, B., Qi, W., Agarwal, T., Lambert, J., Singh, J., Khandelwal, S., Pan, B., Kumar, R., Hartnett, A., Pontes, J.K., Ramanan, D., Carr, P., Hays, J.: Argoverse 2: Next generation datasets for self-driving perception and forecasting. In: Neural Inform. Proc. Syst. Track on Datasets and Benchmarks (2021)
97. Xu, P., Hayet, J.B., Karamouzas, I.: SocialVAE: Human trajectory prediction using timewise latents. In: Eur. Conf. Comput. Vision. pp. 511–528 (2022)
98. Xu, Y., Shao, W., Li, J., Yang, K., Wang, W., Huang, H., Lv, C., Wang, H.: SIND: A drone dataset at signalized intersection in China. In: IEEE Intell. Transport. Syst. Conf. pp. 2471–2478 (2022)
99. Xu, Y., Wang, L., Wang, Y., Fu, Y.: Adaptive trajectory prediction via transferable GNN. In: IEEE/CVF Conf. Comput. Vision Pattern Recog. pp. 6520–6531 (2022)
100. Ye, L., Zhou, Z., Wang, J.: Improving the generalizability of trajectory prediction models with frenet-based domain normalization. In: IEEE Int. Conf. Robot. Automat. pp. 11562–11568. IEEE (2023)
101. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? Advances in Neural Information Processing Systems (NeurIPS) **27** (2014)

102. Yuan, Y., Weng, X., Ou, Y., Kitani, K.M.: Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In: IEEE/CVF Int. Conf. Comput. Vision. pp. 9813–9823 (2021)
103. Zhan, W., Sun, L., Wang, D., Shi, H., Clausse, A., Naumann, M., Kümmerle, J., Königshof, H., Stiller, C., de La Fortelle, A., Tomizuka, M.: INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps. arXiv preprint arXiv:1910.03088 (2019)
104. Zhang, C., Ni, Z., Berger, C.: Spatial-Temporal-Spectral LSTM: A transferable model for pedestrian trajectory prediction. IEEE Trans. Intell. Veh. **9**(1), 2836–2849 (2024)
105. Zhang, Y., Wang, C., Yu, R., Wang, L., Quan, W., Gao, Y., Li, P.: The AD4CHE dataset and its application in typical congestion scenarios of traffic jam pilot systems. IEEE Trans. Intell. Veh. **8**(5), 3312–3323 (2023)
106. Zhao, T., Xu, Y., Monfort, M., Choi, W., Baker, C., Zhao, Y., Wang, Y., Wu, Y.N.: Multi-agent tensor fusion for contextual trajectory prediction. In: IEEE/CVF Conf. Comput. Vision Pattern Recog. pp. 12126–12134 (2019)
107. Zhou, X., Liu, M., Yurtsever, E., Zagar, B.L., Zimmer, W., Cao, H., Knoll, A.C.: Vision language models in autonomous driving: A survey and outlook. IEEE Trans. Intell. Veh. (2024)
108. Zhou, Y., Shao, H., Wang, L., Waslander, S.L., Li, H., Liu, Y.: SmartPretrain: Model-agnostic and dataset-agnostic representation learning for motion prediction. In: Int. Conf. Learning Representations (ICLR) (2025)
109. Zhou, Z., Wang, J., Li, Y.H., Huang, Y.K.: Query-centric trajectory prediction. In: IEEE/CVF Conf. Comput. Vision Pattern Recog. pp. 17863–17873 (2023)