Road Barlow Twins: Redundancy Reduction for Motion Prediction

Royden Wagner[1], Marvin Klemp[1], Carlos Fernandez Lopez[1] and Ömer Şahin Taş[2]
Code: https://github.com/KIT-MRT/road-barlow-twins

***Abstract*— Anticipating the future trajectories of other traffic agents, i.e., motion prediction, is crucial for self-driving vehicles to operate safely in dynamic environments. In this work, we introduce a novel self-supervised pre-training method for motion prediction. Our method is based on Barlow Twins and applies the redundancy reduction principle to embeddings generated from HD maps. Through our method, deep learning models learn augmentation-invariant features of HD maps. We hypothesize that an understanding of the environment can be learned faster using these features. We pre-train several large transformer models and subsequently fine-tune them on motion prediction. Our experiments reveal that the proposed pre-training method can improve mADE and mFDE by 12% and 15% and outperform contrastive learning with PreTraM and SimCLR in a semi-supervised setting.**

I. Introduction

Understanding the trajectories of different traffic agents is critical for self-driving vehicles to operate safely in dynamic environments. Motion prediction in self-driving applications aims to predict the future trajectory of traffic agents based on past trajectories and the given traffic scenario. Recent state-of-the-art methods for motion prediction (e.g., [8], [20], [23]) are deep learning methods trained using supervised learning. As the performance of deep learning methods scales well with the amount of training data [12], [24], [28], there is a great research interest in self-supervised learning methods, which generate supervisory signals from unlabeled data. In the field of computer vision, self-supervised methods are well established (e.g., [5], [9], [21]). For motion prediction in self-driving applications, self-supervised learning is only recently emerging (e.g., [1], [26]). The main reason for this is that until recently the datasets for motion prediction were fewer and considerably smaller (e.g., highD [14] 147 hours recorded vs. Woven Prediction dataset [11] 1001 hours recorded).

In this work, we focus on HD map assisted motion prediction and introduce a novel self-supervised pre-training method. Our hypothesis is that an understanding of the environment can be learned faster when using augmentation-invariant features of HD maps during subsequent fine-tuning. Our method applies the redundancy reduction principle from Barlow Twins [27] to embeddings generated from HD maps. We specifically target transformer models [25] for three reasons: (a) Transformers are flexible foundation models. They are successfully applied to a wide range of applications in natural language processing (e.g., [2], [19], [25]), computer vision (e.g., [3], [6], [18]), and time-series prediction (e.g., [29], [30]). Therefore, it is likely that an improvement in training mechanisms in a particular application will also apply to other applications. (b) Transformers have no inductive biases for generating features based on spatial correlations [22]. Therefore, appropriate mechanisms must be learned from data. (c) The performance of transformers on various downstream tasks scales very well with datasets [12], [28].

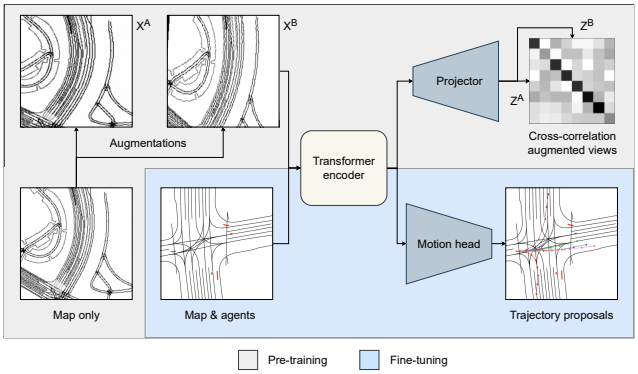

Fig. 1. **Road Barlow Twins model.** During pre-training, plain map data is used. During fine-tuning, annotated samples with past traffic agent trajectories are used. Further details are in Section III-B. Best viewed zoomed-in.

In contrast to related methods, our method does not require annotations of traffic agents [26] and can be trained end-to-end [17]. Overall, our contributions are twofold:

1) We propose and evaluate a novel self-supervised pre-training objective for motion prediction.
2) We introduce two deep learning models for motion prediction, which are based on the foundation model vision transformer.

II. Related work

There are several recent works utilizing self-supervised learning for motion prediction in self-driving. PreTraM [26] exploits for contrastive pre-training that a traffic agent's trajectory is correlated to the map. Inspired by CLIP [21], the similarity of embeddings generated from rasterized HD map images and past agent trajectories is maximised as pre-training. Therefore, past trajectories are required, which limits the application of this method to annotated datasets.

Ma et al. [17] improve modeling interactions between traffic agents via contrastive pre-training with SimCLR [5]. They rasterize images of intersecting agent trajectories and pre-train the corresponding module by maximizing the similarity

[1]The authors are with KIT Karlsruhe Institute of Technology, Engler-Bunte-Ring 21, 76131 Karlsruhe, Germany {firstname.lastname}@kit.edu

[2]Ömer Şahin Taş is with FZI Research Center for Information Technology, Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germany tas@fzi.de

of different views of the same trajectory intersection. Accordingly, only a small part of the motion prediction pipeline is pre-trained and annotations are required to determine the trajectory intersections.

Azevedo et al. [1] use use graph representations of HD maps to generate possible traffic agent trajectories. Trajectories are generated based on synthetic speeds and the connectivity of the graph nodes. The pre-training objective is the same as for the subsequent fine-tuning: motion prediction. While this method is well adapted to motion prediction, it requires non-trivial modeling of agent positions and synthetic velocities when applied to non-annotated data.

Luo et al. [16] fuse information from camera and LiDAR sensors for self-supervised motion prediction. Their method exploits the structural consistency of LiDAR point clouds and cross-sensor motion regularization for motion prediction. Correspondingly, the application of this method requires datasets that contain both camera and LiDAR data.

III. Method

A. Model architectures

In this work, we use a deep learning model similar to MotionCNN [13]. MotionCNN is a baseline model for motion prediction that at the same time achieves competitive performance. Thus, it shares properties with ResNet-50 models [10], which are typically used to evaluate self-supervised methods in computer vision (e.g., [5], [27]). The model is composed of a convolutional neural network (CNN) backbone and a head with a single fully connected layer. As input, rasterized HD map images and rasterized past agent trajectories from the birds-eye view perspective are used. As output, six trajectory proposals and the associated confidences are predicted per traffic agent. All outputs are normalized with the softmax operator. For our first proposed architecture (MotionViT), we replace the CNN backbone in the model architecture with a vision transformer (ViT) [6]. Accordingly, the 224x224 pixel HD map images are split into 16x16 pixel patches and processed by a transformer encoder [25] along positional embeddings and a learned class token (cf. Figure 2). The class token learns a representation of the whole traffic scene as a combination of the patch tokens. Therefore, we use the class token as input for the motion head, which predicts the trajectory proposals.

Inspired by CLIP, we propose a second model architecture (DualMotionViT). The DualMotionViT architecture contains separate transformer encoders for map and agent data. The map and agent tokens, which are generated by the two encoders, are fused in an additional fusion block (cf. Figure 2). We use a memory efficient version of cross-attention [4] to fuse information from both token sets. In detail, the class token from the agent token set attends to the patch tokens of the map token set. Therefore, the computational complexity of the attention mechanism is reduced from $O(n^2)$ to $O(n)$, where n is the number of tokens in either set. Finally, the fused class token serves as input for the motion head.

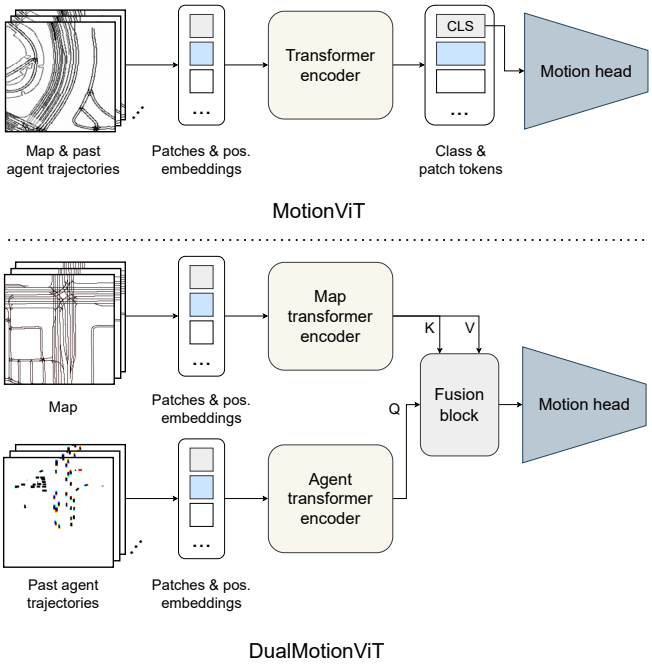

Fig. 2. Proposed model architectures. **MotionViT:** A shared transformer encoder to encode both map data and past agent trajectories. **DualMotionViT:** Separate transformer encoders for map and agent data. Generated embeddings are fused via cross-attention (Queries Q, Keys K, Values V).

B. Pre-training objective

Our proposed pre-training objective, Road Barlow Twins (RBT), is based on Barlow Twins [27]. Barlow Twins is a self-supervised learning method, which aims to learn augmentation-invariant features via redundancy reduction. For each training sample, two views are created by augmentation (X^A and X^B in Figure 1). Afterwards, both views are transformed into feature vectors (Z^A and Z^B) by an encoder model and a projector head. For the two feature vectors, a cross-correlation matrix is created. The pre-training objective is to approximate this cross-correlations matrix to the identity matrix and thus to learn similar feature representations for both views. We adapt this method for motion prediction by pre-training encoder models with map-only data. Compared to annotated samples with map and traffic agent data, these can be created directly from HD maps. During pre-training, our models learn augementation-invariant features from HD maps.

To adapt to the new modality of HD maps, we reduce the strength of pre-training augmentations. We remove gaussian blurring as our models will never process blurred HD map images during fine-tuning or inference for motion prediction. Furthermore, we remove image flipping and limit geometric transformations to moderate rotations (max. +/- 10°) and zooming (max. +/- 30%). The motivation for this is that geometrically highly modified maps should be interpreted differently during the fine-tuning process. Accordingly, the corresponding feature representations should be different as well. Additionally, we use color jitter and color drop as augmentation. Figure 3 shows examples of the augmentations

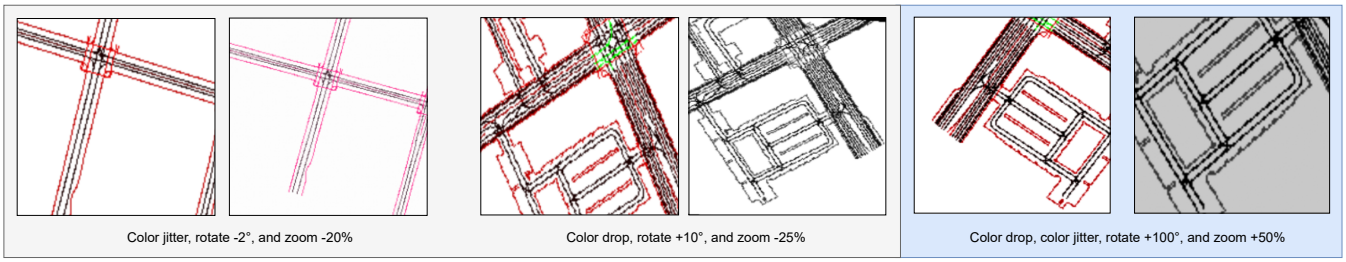

Fig. 3. **Gray:** Proposed weaker augmentations for HD maps. **Blue:** Stronger vanilla Barlow Twins augmentations.

we use and the stronger vanilla Barlow Twins augmentations.

IV. Experiments

A. Choosing a baseline model

To choose a baseline model, we train our model from scratch using a small ViT (ViT-S/16) and a large ViT model (ViT-L/16) as backbones.

Experimental setup. We use a patch size of 16x16 pixels for both variants. For the ViT-S/16 backbone, the first fully connected layer in the motion head has 384 nodes. For the ViT-L/16 backbone, the first fully connected layer in the motion head has 1024 nodes. For both backbones the second fully connected layer has six nodes to predict six trajectory proposals. We train all models with AdamW [15] as optimizer for 190 epochs. The initial learning rate is set to 10^{-4} and reduced to 10^{-6} using a cosine annealing learning rate scheduler. As loss, we are minimizing the negative multivariate log-likelihood loss. We use a global batch size of 512 for the ViT-S/16 model and 384 for the ViT-L/16 model. We train in a data-distributed-parallel (ddp) manner with per GPU batch sizes of 128 for the ViT-S/16 models and 96 for the ViT-L/16 models.

Dataset. We use the official training and validation splits of the Waymo Open Motion [7] dataset as training and validation data. Correspondingly, we train with 2.2M training samples. Validation and evaluation are performed on 200K samples. In addition to the map, the trajectories of all traffic agents from the last second are used as input during fine-tuning for motion prediction. The data set is sampled with 10 Hz, accordingly 10 time points are used as input.

Evaluation metrics. We use the negative multivariate log-likelyhood (NLL), the average displacement error (ADE), the final displacement error (FDE), and mean squared error (MSE) to evaluate the trajectory proposals. The ADE and FDE scores are evaluated in the oracle/minimum mode. Accordingly, the distance errors of the trajectory proposal with the lowest distance error are measured. Following [7], all metrics are computed at different prediction horizons of 3s and 5s and averaged.

Results. Table I shows the achieved evaluation metrics. The larger ViT-L model consistantly yields better (lower) scores for all considered metrics. Therefore, we choose this model as baseline in the following experiments.

TABLE I
Scaling ViT backbones to choose a baseline model

Backbone	NLL ↓	mADE ↓	mFDE ↓	MSE ↓	#Params
ViT-S	64.4	0.897	2.069	1.852	24.2M
ViT-L	49.6	0.794	1.865	1.134	310M

B. Comparing pre-training methods

In this set of experiments, we compare our proposed pre-training method with contrastive learning. As examples for contrastive learning, we use two configurations of PreTraM and the vanilla SimCLR. The first configuration of PreTraM is a combination of map contrastive learning (MCL) via SimCLR and trajectory-map contrastive learing (TMCL). For TMCL, the similarity of embeddings from map data and past agent trajectories of the same scene is maximized. As second configuration of PreTraM, we evaluate TMCL without MCL. Following MotionCNN, which is pre-trained on ImageNet, we additionally evaluate a combination of our method and ImageNet pre-training. In this case, we initialize the ViT-L model with ImageNet weights before pre-training with our method. This approach can be seen as a form of two-stage pre-training.

Dataset. Initially, we use the same dataset configuration as in the previous experiment. Afterwards, we reduce the number of training sample to a fraction of 30% (f_{ds}).

Evaluation metrics. In addition to the mADE and mFDE scores, we introduce the $\Delta\text{mADE}_{\text{rel}}$ and $\Delta\text{mADE}_{\text{rel}}$ scores. $\Delta\text{mADE}_{\text{rel}}$ and $\Delta\text{mFDE}_{\text{rel}}$ measure the relative change w.r.t. the baselines without pre-training. T_{rel} measures the relative change in training time w.r.t. the baselines.

Experimental setup. For fine-tuning on motion prediction, we use the same experimental setup as in the previous experiment. For TMCL pre-training, we use the fine-tuning dataset without trajectory labels. For the RBT, MCL, and SimCLR pre-training, we use the same dataset, but remove all data related to traffic agents. For all three methods and backbone models, we add a projection head with 3 fully connected layers. The first layer has 1024 nodes, the following two layers have 2048 nodes. The training samples are augmented using the weaker augmentations shown in Figure 2. For the DualMotionViT models, we use two ViT-B/16 models as map and agent encoders.

Results. Table II shows the results of this experiment.

TABLE II

COMPARING PRE-TRAINING METHODS USING DIFFERENT MODELS AND DATASET SIZES

Model	Pre-training	Config	f_{ds}	NLL ↓	mADE ↓	ΔmADE$_{\text{rel}}$	mFDE ↓	ΔmFDE$_{\text{rel}}$	T_{rel}	#Params
MotionViT	None		100%	49.6	0.794		1.865		1.0	310M
	RBT (Our)		100%	**39.2**	**0.697**	-12.22%	**1.584**	-15.07%	2.0	310M
	SimCLR		100%	76.4	1.172	+47.61%	2.483	+33.14%	2.0	310M
	RBT	+ ImageNet	100%	65.1	0.942	+18.64%	2.297	+23.16%	2.0	310M
MotionViT	None		30%	**87.3**	1.096		2.554		1.0	310M
	RBT (Our)		30%	87.8	**1.089**	-0.64%	**2.552**	-0.08%	2.0	310M
	SimCLR		30%	434.9	2.921	+166.51%	11.494	+350.04%	2.0	310M
	RBT	+ ImageNet	30%	87.8	1.107	+1.00%	2.602	+1.88%	2.0	310M
DualMotionViT	None		100%	43.7	0.723		1.672		1.0	190M
	RBT (Our)		100%	38.7	0.691	-4.43%	1.565	-6.40%	2.0	190M
	PreTraM	MCL + TMCL	100%	45.6	0.718	-0.69%	1.643	-1.73%	2.0	190M
	PreTraM	TMCL	100%	**36.0**	**0.679**	-6.09%	**1.529**	-8.55%	2.0	190M

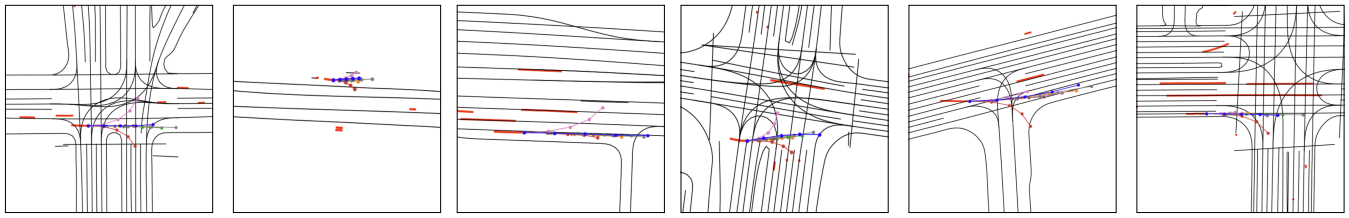

Fig. 4. **Trajectory proposals of our MotionViT model.** Past trajectories of traffic agents are visualized as red bars, the ground truth future trajectory as blue lines and the trajectory proposals of our model are in decreasing order of confidence scores: orange, green, red, brown, pink and grey. Discrete predicted agent locations are marked as dots.

When fine-tuning on the complete training dataset, our pre-training method improves the mADE score by 12% and the mFDE score by 15% compared to the baseline without pre-training. Contrastive pre-training performs worst in this setup. We hypothesize that this due to the fact that augmented views of different HD maps are still rather similar to each other compared to images of different classes in ImageNet (e.g., cars and birds). During contrastive pre-training, all samples in a batch other than the current one are treated as negative examples. Therefore, the pre-training objective becomes to learn dissimilar embeddings for rather similar samples. Initializing the model with weights learned form training on ImageNet worsens the performance as well. This highlights the domain gap between general purpose vision datasets and motion prediction datasets with HD maps. This gap also seems to have a stronger impact on our architecture than, for example, on MotionCNN. One possible interpretation is that the learned feature extraction mechanisms in a ViT adapt more to the training data than in CNNs [22]. Fine-tuning on 30% of the training split yields similar results, but the differences in performance are less significant and all methods perform worse. This could be due to the fact that our pre-training method learns useful feature extraction mechanisms only for environment perception. However, for motion prediction, the information about traffic agents is crucial. This needs to be learned by a large transformer model, for which sufficient data is needed.

The third block in Table II shows the results achieved with the DualMotionViT model. Already without pre-training, 10% lower mADE and 9% lower mFDE scores are achieved than with the MotionViT model. The DualMotionViT model also requires 120M fewer parameters to achieve these scores. This shows that it is advantageous for motion prediction to first process map and agent data individually and then merge them in the embedding space. Our pre-training method can reduce the mADE and mFDE scores by a further 4% and 6% respectively. Vanilla PreTraM with MCL and TMCL improves the mADE and mFDE scores only by 1% and 2%. Therefore, our method can outperform vanilla PreTraM in this semi-supervised setting. Overall, however, our new variation of PreTraM without MCL performs best in this experiment.

In real-world scenarios, the usefulness of motion prediction methods can also be assessed based on the plausibility of the multiple trajectory proposals. Therefore, we show qualitative results of our MotionViT model (ViT-L/16 pre-trained on a full dataset with RBT) in Figure 4. Overall, our model is able to predict a diverse set of possible future trajectories in different traffic scenarios.

V. CONCLUSION

We have introduced a novel self-supervised pre-training method for motion prediction in self-driving applications. The proposed method builds up-on Barlow Twins and learns augmentation-invariant features of HD maps. In contrast to related methods, our method does not require annotations of traffic agents and can be trained end-to-end. Our experiments revealed that our pre-training method can improve the accuracy of motion prediction and outperform contrastive learning. Furthermore, we proposed two transformer-based baseline models for motion prediction. Future steps include evaluating if our pre-training method can be extended to graph representations of HD maps, which have a higher information density and are more memory efficient.

ACKNOWLEDGMENT

This research is accomplished within the project HAIBrid (FKZ 01IS21096A). We acknowledge the financial support for the project by the Federal Ministry of Education and Research of Germany (BMBF). Furthermore, this work was supported by the Helmholtz Association's Initiative and Networking Fund on the HAICORE@FZJ partition.

REFERENCES

[1] Caio Azevedo, Thomas Gilles, Stefano Sabatini, and Dzmitry Tsishkou. Exploiting map information for self-supervised learning in motion forecasting. *arXiv preprint arXiv:2210.04672*, 2022.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020.

[4] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, 2021.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

[7] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9710–9719, 2021.

[8] Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15303–15312, 2021.

[9] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] John Houston, Guido Zuidhof, Luca Bergamini, Yawei Ye, Long Chen, Ashesh Jain, Sammy Omari, Vladimir Iglovikov, and Peter Ondruska. One thousand and one hours: Self-driving motion prediction dataset. In *Conference on Robot Learning*, pages 409–418. PMLR, 2021.

[12] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[13] Stepan Konev, Kirill Brodt, and Artsiom Sanakoyeu. Motioncnn: a strong baseline for motion prediction in autonomous driving. *arXiv preprint arXiv:2206.02163*, 2022.

[14] Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2118–2125. IEEE, 2018.

[15] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[16] Chenxu Luo, Xiaodong Yang, and Alan Yuille. Self-supervised pillar motion learning for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3183–3192, 2021.

[17] Hengbo Ma, Yaofeng Sun, Jiachen Li, and Masayoshi Tomizuka. Multi-agent driving behavior prediction across different scenarios with self-supervised domain knowledge. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3122–3129. IEEE, 2021.

[18] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8844–8854, 2022.

[19] OpenAI. Gpt-4 technical report, 2023.

[20] Stefano Pini, Christian S Perone, Aayush Ahuja, Ana Sofia Rufino Ferreira, Moritz Niendorf, and Sergey Zagoruyko. Safe real-world autonomous driving by learning to predict and plan with a mixture of experts. *arXiv preprint arXiv:2211.02131*, 2022.

[21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[22] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34:12116–12128, 2021.

[23] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Motion transformer with global intention localization and local movement refinement. In *36th Conference on Neural Information Processing Systems*, 2022.

[24] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[26] Chenfeng Xu, Tian Li, Chen Tang, Lingfeng Sun, Kurt Keutzer, Masayoshi Tomizuka, Alireza Fathi, and Wei Zhan. Pretram: Self-supervised pre-training via connecting trajectory and map. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, pages 34–50. Springer, 2022.

[27] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

[28] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022.

[29] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.

[30] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, pages 27268–27286. PMLR, 2022.