Supplementary Material for SSL-Lanes: Self-Supervised Learning for Motion Forecasting in Autonomous Driving

1 Detailed Network Architecture for Baseline

We provide the detailed network architecture of our baseline in this section and is illustrated in Fig. 1.

For the *agent feature extractor*, the architecture is similar to [1]. We use an 1D CNN to process the trajectory input. The output is a temporal feature map, whose element at $t = 0$ is used as the agent feature. The network has three groups/scales of 1D convolutions. Each group consists of two residual blocks [2], with the stride of the first block as 2. Feature Pyramid Network (FPN) [3] fuses the multi-scale features, and applies another residual block to obtain the output tensor. For all layers, the convolution kernel size is 3 and the number of output channels is 128. Layer normalization [4] and Rectified Linear Unit (ReLU) are used after each convolution.

The *map feature extractor* has two LaneConv residual [2] blocks which are the stack of a LaneConv(1, 2, 4, 8, 16, 32) and a linear layer, as well as a shortcut. All layers have 128 feature channels. Layer normalization [4] and ReLU are used after each LaneConv and linear layer.

For the map-aware agent feature (M2A) module, the distance threshold is 12m. It is 100m for the agent-to-agent (A2A) interaction module. The two *interaction modules* have two residual blocks, which consist of a stack of an attention layer and a linear layer, as well as a residual connection. All layers have 128 output feature channels.

Taking the interaction-aware actor features as input, our *trajectory decoder* is a multi-modal prediction header that outputs the final motion forecasting. For each agent, it predicts K possible future trajectories and confidence scores. The header has two branches, a regression branch to predict the trajectory of each mode and a classification branch to predict the confidence score of each mode.

Key differences with Lane-GCN [1]: Our main difference is we use two Lane-Conv blocks instead of four as map-feature extractor in order to prevent over-smoothing in GNNs [5]. We also do not use the four-way fusion proposed by Lane-GCN and do away with the agent to map (A2M) and the map to map (M2M) interaction blocks, which saves compute and memory.

2 SSL-Lanes: Self-Supervision meets Motion Forecasting

Before we discuss designing pretext tasks to generate self-supervisory signals, we consider a scheme that will allow combined training for self-supervised pretext tasks and our standard framework.

How to combine motion forecasting and SSL? Self-supervision can be combined with motion forecasting in various ways. In one scheme we could pre-train the forecasting encoder with pretext tasks (which can be viewed as an initialization for the encoder's parameters) and then fine-tune the pre-trained encoder with a downstream decoder. In another scheme, we could choose to freeze the encoder and only train the decoder. In a third scheme, we could optimize our pretext task and primary task *jointly*, as a kind of multi-task learning setup. Inspired by relevant discussions in GNNs, we choose the third-scheme, i.e., multi-task learning, which is the most general framework among the three and is also experimentally verified to be the most effective [6, 7].

Joint Training: Considering our motion forecasting task and a self-supervised task, the output and the training process can be formulated as:

$$\mathbf{\Psi}^\star, \mathbf{\Omega}^\star, \mathbf{\Theta}_{\text{ss}}^\star = \underset{\mathbf{\Psi}, \mathbf{\Omega}, \mathbf{\Theta}_{\text{ss}}}{\arg\min} \quad \alpha_1 \mathcal{L}_{\text{sup}}(\mathbf{\Psi}, \mathbf{\Omega}) + \alpha_2 \mathcal{L}_{\text{ss}}(\mathbf{\Psi}, \mathbf{\Theta}_{\text{ss}}) \tag{1}$$

where, $\mathcal{L}_{\text{ss}}(\cdot, \cdot)$ is the loss function of the self-supervised task, $\mathbf{\Theta}_{\text{ss}}$ is the corresponding linear transformation parameter, and $\alpha_1, \alpha_2 \in \mathrm{R}_{>0}$ are the weights for the supervised and self-supervised

6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.

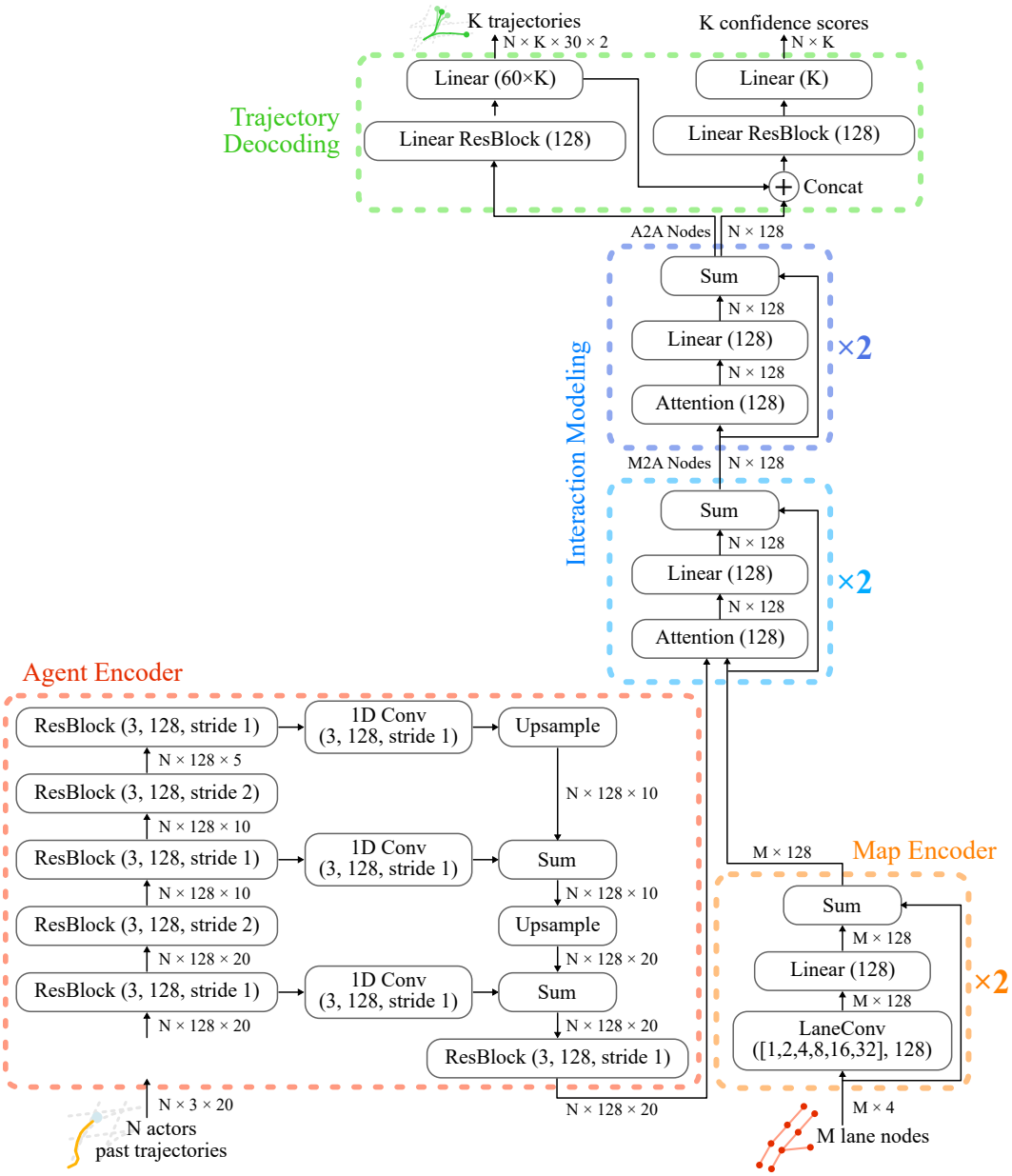

Figure 1: Architecture of the baseline model

losses. If the pretext task only focuses on the map encoder, then $\mathbf{\Psi} = \{\mathbf{\Theta}\}$ and $\mathbf{\Omega} = \{g_{\text{enc}}, \mathbf{\Lambda}, g_{\text{dec}}\}$. Otherwise, $\mathbf{\Psi} = \{g_{\text{enc}}, \mathbf{\Theta}, \mathbf{\Lambda}\}$ and $\mathbf{\Omega} = \{g_{\text{dec}}\}$. Henceforth, we also define the following representations. We will represent the primary task encoder as function $f_{\mathbf{\Psi}}$, parameterized by $\mathbf{\Psi}$. Furthermore, given a pretext task, which we will design in the next section, the pretext decoder $p_{\mathbf{\Theta}_{\text{ss}}}$ is a function that predicts pseudo-labels and is parameterized by $\mathbf{\Theta}_{\text{ss}}$.

Benefit of SSL-Lanes: In Eq. (1), the self-supervised task as a regularization term throughout network training. It acts as the regularizer learned from unlabeled data under the minor guidance of human prior (design of pretext task). Therefore, a properly designed task would introduce data-driven prior knowledge that improves model generalizability.

3 Pretext tasks for Motion Forecasting

At the core of our SSL-Lanes approach is defining pretext tasks based upon self-supervised information from the underlying map structure *and* the overall temporal prediction problem itself. Our core approach is simple in contrast to state-of-the-art that rely on complex encoding architectures [8, 9, 10, 11, 1, 12, 13], ensembling forecasting heads [14, 15], involved final goal-set optimization algorithms [16, 17] or heavy fusion mechanisms [1], to improve prediction performance.

3.1 Lane-Masking

Motivation: The goal of the *Lane-Masking* pretext task is to encourage the map encoder $\mathbf{\Psi} = \{\mathbf{\Theta}\}$ to learn local structure information in addition to the forecasting task that is being optimized. In this task, we learn by recovering feature information from the perturbed lane graphs.

Benefit of Lane-Masking: Since Argoverse [18] has imbalanced data with respect to maneuvers, there are cases when right/left turns, lane-changes, acceleration/deceleration are missed by the baseline even with multi-modal predictions. We hypothesize that stronger map-features can help the multi-modal prediction header to infer that some of the predictions should also be aligned with map topology. For example, even if an agent is likely to go straight at an intersection, some of the possible futures should also cover acceleration/deceleration or right/left turns guided by the local map structure.

3.2 Distance to Intersection

Motivation: The Lane-Masking pretext task is from a local structure perspective based on masking and trying to predict local attributes of the vectorized HD-map. We further develop the *Distance-to-Intersection* pretext task to guide the map-encoder, $\mathbf{\Psi} = \{\mathbf{\Theta}\}$, to maintain global topology information by predicting the distance (in terms of shortest path length) from all lane nodes to intersection nodes. Datasets like Argoverse [18] provide lane attributes which describe whether a lane node is located within an intersection. This will force the representations to learn a global positioning vector of each of the lane nodes.

Benefit of Distance to Intersection Task: We hypothesize that since change of speed, acceleration, primary direction of movement etc. for an agent can change far more dramatically as an agent approaches or moves away from an intersection, it is beneficial to explicitly incentivize the model to pick up the geometric structure near an intersection and compress the space of possible map-feature encoders, thereby effectively simplifying inference. We also expect this to improve drivable area compliance nearby an intersection, which is often a problem for current motion forecasting models.

3.3 Maneuver Classification

Motivation: The Lane-Masking and Distance to Intersection pretext tasks are both based on extracting feature and topology information from a HD-map. However, pretext tasks can also be constructed from the overall forecasting task itself. Thus we propose to obtain free pseudo-labels in the form of a 'maneuver' the agent-of-interest intends to execute, and define a set of 'intentions' to represent common semantic modes (e.g. change lane, speed up, slow down, turn-right, turn-left etc.) We call this pretext task *Maneuver Classification*, and we expect it to provide prior regularization to $\mathbf{\Psi} = \{g_{\text{enc}}, \mathbf{\Theta}, \mathbf{\Lambda}\}$, based on driving modes.

Benefit of Maneuver Classification Task: We hypothesize if one can identify the intention of a driver, the future motion of the vehicle will match that maneuver, thereby reducing the set of possible end-points for the agent. We also expect that agents with similar maneuvers will tend to have consistent semantic representations.

3.4 Forecasting Success/Failure Classification

Motivation: In contrast to maneuver classification, which provides coarse-grained prediction of the future, self-supervision mechanisms can also offer a strong learning signal through goal-reaching tasks which are generated from the agent's trajectories. We propose a pretext task called *Success/Failure Classification*, which trains an agent specialized at achieving end-point goals which directly lead to the forecasting-task solution. We expect this to constrain $\mathbf{\Psi} = \{g_{\text{enc}}, \mathbf{\Theta}, \mathbf{\Lambda}\}$ to predict trajectories ϵ distance away from the correct final end-point. Conceptually, the more examples of successful goal states we collect, the better understanding of the target goal of the forecasting task we have.

Benefit of Success/Failure Classification Task: We hypothesize that this task will especially provide stronger gains for cases where the final end-point is not aligned with the general direction of agent movement for majority of samples given in the dataset, and is thus not well captured by average displacement based supervised loss functions.

4 Discussion: SSL-Lanes vs. State-of-the-Art

We use this section to distinguish our work from methods that we believe have similar intuition but very different construction, in order to highlight its novelty and value.

- SSL-Lanes vs. VectorNet [19]: Vector-Net is the only other motion forecasting work that proposes to randomly mask out the input node features belonging to either scene context or agent trajectories, and ask the model to reconstruct the masked features. Their intuition is to encourage the graph networks to better capture the interactions between agent dynamics and scene context. However, our motivation differs from VectorNet in two respects: (a) We propose to use masking to learn local map-structure better, as opposed to learning interactions between map and the agent. This is an easier optimization task, and we outperform VectorNet. (b) A lane is made up of several nodes. We propose to randomly mask out a certain percentage of each lane. This is a much stronger prior as compared to randomly masking out any node (which may correspond to either a moving agent or map) and ensures that the model pays attention to all parts of the map.
- SSL-Lanes vs. CS-LSTM [20]: CS-LSTM appends the encoder context vector with a one-hot vector corresponding to the lateral maneuver class and a one-hot vector corresponding to the longitudinal maneuver class. Subsequently, the added maneuver context allows the decoder LSTM to generate maneuver specific probability distributions. This construction however is quite different from our work because it is not auxiliary in nature - it always outputs and appends a maneuver to the decoder, even during inference. This we believe is too strong of a bias for the prediction model, especially given the fact that the maneuvers are generated using very simple velocity profiles and not from careful mining of the data. In our conditioning, the maneuvers are mined from data and the final motion prediction does not depend directly on them. We believe this design is much more flexible since it allows to generate more supervisory signals in the form of maneuvers during training, but at the same time does not require an explicit maneuver to condition the final future forecast trajectory output during inference.
- SSL-Lanes vs. MultiPath [21]: MultiPath is also not auxiliary in nature: it factorizes motion uncertainty into intent uncertainty and control uncertainty; models the uncertainty over a discrete set of intents with a softmax distribution; and then outputs control uncertainty as a Gaussian distribution dependent on each waypoint state of the anchor trajectory (corresponding to the intent). While this construction is highly intuitive and effective by design, it is very different from our SSL-based construction. Ours is an auxiliary task which provides supervision during training, and effectively functions as a regularizer, while being general enough to be used with any other data-driven motion forecasting model.

5 Discussion: Choice of Dataset

We now compare the commonly used motion-forecasting datasets, i.e., nuScenes [22], Waymo-Open-Motion-Dataset (WOMD) [23] and Argoverse [18]. We individually discuss why Argoverse is best positioned to bring out the benefits of our proposed work.

- *Scale of Data:* We first compare the dataset size. We note that Argoverse is not only two orders larger than nuScenes, and also has greater number of training samples and unique trajectories compared to WOMD.

	nuScenes	WOMD	Argoverse
Number of Unique Tracks:	4.3k	7.65m	11.7m
Number of Training Segments:	1k	104k	324k

- *Interesting Scenarios for Forecasting Evaluation:* We next compare if the datasets specifically mines for interesting scenarios, which is the area we want to improve the current baseline. nuScenes was not collected to capture a wide diversity of complex and interesting driving scenarios. WOMD on the other hand specifically mines for pairwise interaction scenarios, where the main objective is to improve forecasting for interacting agents. However, the scope of our study is to primarily focus on motion at intersections undergoing lane-changes and turns. We expect the SSL-losses to improve understanding of the context/environment, trajectory embeddings and address data-imbalance w.r.t. maneuvers. We leave heavy interaction-based use cases for future work. Finally, Argoverse mines for interesting motion patterns at intersections, which involve lane-changes, acceleration/deceleration, and turns. We thus find this dataset best suited to showcase our proposed method.
- *Community focus on Argoverse:* We also find that many popular motion forecasting methods published by the robotics community have also included evaluations only on the Argoverse dataset including: Lane-GCN, Lane-RCNN, PRIME, DCMS, TPCN, mm-Transformer, HiVT, Multi-modal Transformer, DSP etc. This makes it easier for us to position our work with respect to these approaches.

6 Implementation of Pretext Tasks

In this section, we discuss various design decisions for the proposed pretext tasks.

6.1 Lane-Masking

For this pretext task, we mask m_a percent of every lane and reconstruct its features. In Tab. 1, we

Method	$\mathbf{m}_a$	**minADE**$_6$	**minFDE**$_6$	**MR**$_6$
Baseline	-	0.73	1.12	11.07
Random Masking	0.4	0.71	1.03	9.11
Lane-Masking	0.3	0.71	1.04	9.02
Lane-Masking	0.4	**0.70**	**1.02**	**8.84**
Lane-Masking	0.5	0.71	1.05	9.31

Table 1: Effect of masking ratio (m_a) on forecasting performance for lane-masking task

study the influence of masking ratio on the final forecasting performance. Random masking refers to masking out m_a percent random map nodes and lane-masking refers to masking out m_a percent of lanes in the map. We finally choose $m_a = 0.4$ as the most effective parameter for the lane-masking pretext task, which outperforms random masking. The model infers missing lane-nodes to produce plausible outputs during reconstruction. We hypothesize that this reasoning is linked to learning useful representations.

6.2 Distance to Intersection

For this pretext task, we explore two different options for framing the problem of predicting the distance to the nearest intersection node in Tab. 2. We first explore predicting this distance as a classification task. We group the lengths into four categories: $d_{ij} = 1$, $d_{ij} = 2$, $d_{ij} = 3$, $d_{ij} = 4$ and $d_{ij} >= 5$. We however find that this is harder to optimize than the regression loss proposed in our methods section, which we finally choose as our loss for the distance to intersection pretext task.

Method	Pretext Loss	minADE_6	minFDE_6	MR_6
Baseline	-	0.73	1.12	11.07
Distance to Intersection	Classification	0.72	1.06	9.64
Distance to Intersection	Regression	**0.71**	**1.04**	**8.93**

Table 2: Effect of pretext loss type on forecasting performance for distance to intersection task

6.3 Maneuver Classification

For this pretext task, we first divide the lateral and longitudinal maneuvers by choosing a threshold angle of 20 from the vertical. We next find that constrained k-means [24] on agent end-points for lateral and longitudinal maneuvers works best to separate the trajectory samples into different clusters. This is illustrated in Fig. 2. For differentiating the longitudinal maneuvers from the lane-change maneuver, we check a combination of the distance from the lane centerlines for start and stop positions and the orientations of the nearest centerline for start and stop positions.

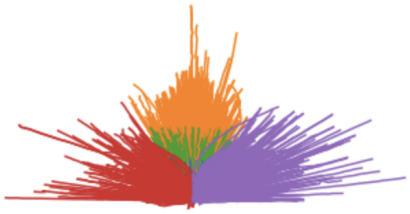

Figure 2: Modes of driving from unsupervised clustering of data

6.4 Success/Failure Classification

For this pretext task, the primary bottleneck is the fact that the number of positive examples if far fewer than the number of negative examples. This is because there are only a few success examples in a 2m area near the end-point of a single recorded ground-truth trajectory, while the rest of the points in the scene can be considered as failure examples. We consider first setting $\epsilon = 3$m, i.e. a wider area for success examples, and then reducing it to $\epsilon = 2$m linearly over the total number of training steps. We find that this can actually harm the final forecasting performance. We thus follow [25] to use focal loss to train our auxiliary classification task.

7 Similarity in feature space

We analyze the CKA similarity [26] between the representations learnt by: a model trained with pretext task 'D2I' (refers to distance to intersection task) and baseline; two models trained with different pretext tasks. In Fig. 3, Base(M2A) refers to $\tilde{\boldsymbol{p}}_i$, Base(A2A) refers to $\acute{\boldsymbol{p}}_i$. 'Mask' refers to lane-masking, 'success/fail' refers to success or failure classification task and 'intention' suggests maneuver classification.

Our main questions are: (a) how much does the pretext task feature differ from the baseline? (b) do the features from different pretext tasks collapse to the same feature? First we note that representation learned by D2I does not *collapse* to the same representation learned by Mask or Success/Fail or Intention. Secondly we note that D2I features are quite different from Base-M2A features $\tilde{\boldsymbol{p}}_i$

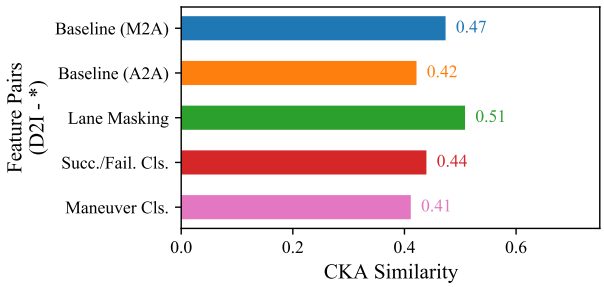

Figure 3: CKA Feature similarity between feature pairs of baseline and different pretext tasks. Similarity score is 1 for completely overlapping features and 0 for completely divergent features.

and Base-A2A features $\acute{\boldsymbol{p}}_i$, which suggests that *task-specific regularization* has indeed resulted in different parameters.

8 Metrics

ADE is defined as the average displacement error between ground-truth trajectories and predicted trajectories over all time steps. FDE is defined as displacement error between ground-truth trajectories and predicted trajectories at the final time step. We compute K likely trajectories for each scenario with the ground truth label, where $K = 1$ and $K = 6$ are used. Therefore, minADE and minFDE are minimum ADE and FDE over the top K predictions, respectively. Miss rate (MR) is defined as the percentage of the best-predicted trajectories whose FDE is within a threshold (2 m). Brier-minFDE is the minFDE plus $(1 - p)^2$, where p is the corresponding trajectory probability.

9 Qualitative Results

We next present some multi-modal prediction trajectories on several hard cases shown in Fig. 4. SSL-Lanes can capture left and right turns better, while also being able to discern acceleration at intersections. Our pretext tasks provide priors for the model and provides data-driven regularization for free. This can improve forecasting because of better understanding of map topology, agent context with respect to the map, and also improve generalization for maneuver imbalance implicitly present in data.

10 Discussion: Potential of this Work

We expect this work to influence real world deployment of SSL forecasting methods for autonomous driving. Another use case for this work is realistic behavior generation in traffic simulation. The general construction of the prediction problem, inspired by [1], enables a generic understanding of how an object moves in a given environment without memorizing the training data. A neural network may learn to associate particular areas of a scene with certain motion patterns. To prevent this, we centre around the agent of interest and normalize all other trajectory and map coordinates with respect to it. We predict relative motion as opposed to absolute motion for the future trajectory. This helps to learn general motion patterns. Reconstructing the map or predicting distances from map elements are conducted in a frame-of-reference relative to the agent of interest. This helps in learning general map connectivity. Following work in pedestrian trajectory prediction, we also additionally add random rotations to the training trajectories to reduce directional bias. Furthermore, we provide strong evidence that SSL-based tasks provide better generalization compared to pure supervised training, thereby having the ability to effectively reuse the same prediction model across different scenarios.

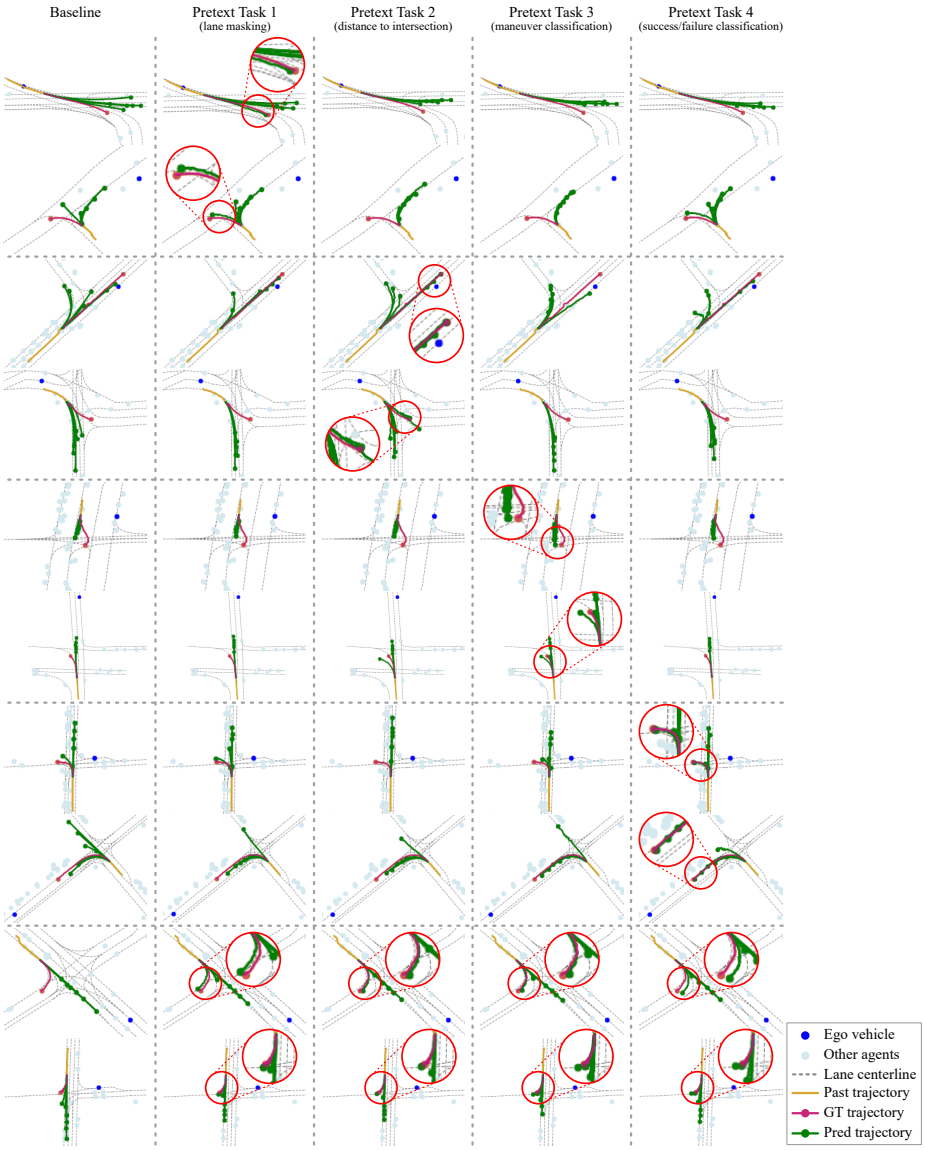

Figure 4: Visual results of our proposed SSL-Lanes on the Argoverse [18] validation set. Generally, these qualitative results demonstrate the effectiveness of our proposed pretext tasks.

References

[1] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun. Learning lane graph representations for motion forecasting. In *ECCV*, 2020. 1, 3, 7

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. doi:10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90. 1

[3] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 936–944. IEEE Computer Society, 2017. doi:10.1109/CVPR.2017.106. URL https://doi.org/10.1109/CVPR.2017.106. 1

[4] L. J. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. URL http://arxiv.org/abs/1607.06450. 1

[5] Q. Li, Z. Han, and X. Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In S. A. McIlraith and K. Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3538–3545. AAAI Press, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16098. 1

[6] Y. You, T. Chen, Z. Wang, and Y. Shen. When does self-supervision help graph convolutional networks? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 10871–10880. PMLR, 2020. URL http://proceedings.mlr.press/v119/you20a.html. 1

[7] W. Jin, T. Derr, H. Liu, Y. Wang, S. Wang, Z. Liu, and J. Tang. Self-supervised learning on graphs: Deep insights and new direction. *CoRR*, abs/2006.10141, 2020. URL https://arxiv.org/abs/2006.10141. 1

[8] W. Zeng, M. Liang, R. Liao, and R. Urtasun. Lanercnn: Distributed representations for graph-centric motion forecasting. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2021, Prague, Czech Republic, September 27 - Oct. 1, 2021*, pages 532–539. IEEE, 2021. doi:10.1109/IROS51168.2021.9636035. URL https://doi.org/10.1109/IROS51168.2021.9636035. 3

[9] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou. Multimodal motion prediction with stacked transformers. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 7577–7586. Computer Vision Foundation / IEEE, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Liu_Multimodal_Motion_Prediction_With_Stacked_Transformers_CVPR_2021_paper.html. 3

[10] J. Ngiam, B. Caine, V. Vasudevan, Z. Zhang, H. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal, D. Weiss, B. Sapp, Z. Chen, and J. Shlens. Scene transformer: A unified multi-task model for behavior prediction and planning. *CoRR*, abs/2106.08417, 2021. URL https://arxiv.org/abs/2106.08417. 3

[11] S. Khandelwal, W. Qi, J. Singh, A. Hartnett, and D. Ramanan. What-if motion prediction for autonomous driving. *CoRR*, abs/2008.10587, 2020. URL https://arxiv.org/abs/2008.10587. 3

[12] Z. Huang, X. Mo, and C. Lv. Multi-modal motion prediction with transformer-based neural network for autonomous driving. *CoRR*, abs/2109.06446, 2021. URL https://arxiv.org/abs/2109.06446. 3

[13] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, C. Li, and D. Anguelov. TNT: target-driven trajectory prediction. In J. Kober, F. Ramos, and C. J. Tomlin, editors, *4th Conference on Robot Learning, CoRL 2020, 16-18 November 2020, Virtual Event / Cambridge, MA, USA*, volume 155 of *Proceedings of Machine Learning Research*, pages 895–904. PMLR, 2020. URL https://proceedings.mlr.press/v155/zhao21b.html. 3

[14] M. Ye, J. Xu, X. Xu, T. Cao, and Q. Chen. DCMS: motion forecasting with dual consistency and multi-pseudo-target supervision. *CoRR*, abs/2204.05859, 2022. doi:10.48550/arXiv.2204.05859. URL https://doi.org/10.48550/arXiv.2204.05859. 3

[15] B. Varadarajan, A. Hefny, A. Srivastava, K. S. Refaat, N. Nayakanti, A. Cornman, K. Chen, B. Douillard, C. Lam, D. Anguelov, and B. Sapp. Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction. *CoRR*, abs/2111.14973, 2021. URL https://arxiv.org/abs/2111.14973. 3

[16] J. Gu, C. Sun, and H. Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 15283–15292. IEEE, 2021. doi:10.1109/ICCV48922.2021.01502. URL https://doi.org/10.1109/ICCV48922.2021.01502. 3

[17] H. Song, D. Luan, W. Ding, M. Y. Wang, and Q. Chen. Learning to predict vehicle trajectories with model-based planning. In A. Faust, D. Hsu, and G. Neumann, editors, *Conference on Robot Learning, 8-11 November 2021, London, UK*, volume 164 of *Proceedings of Machine Learning Research*, pages 1035–1045. PMLR, 2021. URL https://proceedings.mlr.press/v164/song22a.html. 3

[18] M. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays. Argoverse: 3d tracking and forecasting with rich maps. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8748–8757. Computer Vision Foundation / IEEE, 2019. doi:10.1109/CVPR.2019.00895. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Chang_Argoverse_3D_Tracking_and_Forecasting_With_Rich_Maps_CVPR_2019_paper.html. 3, 5, 8

[19] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid. Vectornet: Encoding HD maps and agent dynamics from vectorized representation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 11522–11530. Computer Vision Foundation / IEEE, 2020. doi:10.1109/CVPR42600.2020.01154. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Gao_VectorNet_Encoding_HD_Maps_and_Agent_Dynamics_From_Vectorized_Representation_CVPR_2020_paper.html. 4

[20] N. Deo and M. M. Trivedi. Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms. In *2018 IEEE Intelligent Vehicles Symposium, IV 2018, Changshu, Suzhou, China, June 26-30, 2018*, pages 1179–1184. IEEE, 2018. doi:10.1109/IVS.2018.8500493. URL https://doi.org/10.1109/IVS.2018.8500493. 4

[21] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In L. P. Kaelbling, D. Kragic, and K. Sugiura, editors, *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pages 86–99. PMLR, 2019. URL http://proceedings.mlr.press/v100/chai20a.html. 4

[22] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 11618–11628. Computer Vision Foundation / IEEE, 2020. doi:10.1109/CVPR42600.2020.01164. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Caesar_nuScenes_A_Multimodal_Dataset_for_Autonomous_Driving_CVPR_2020_paper.html. 5

[23] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou, Z. Yang, A. Chouard, P. Sun, J. Ngiam, V. Vasudevan, A. McCauley, J. Shlens, and D. Anguelov. Large scale interactive motion forecasting for autonomous driving : The waymo open motion dataset. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9690–9699. IEEE, 2021. doi:10.1109/ICCV48922.2021.00957. URL https://doi.org/10.1109/ICCV48922.2021.00957. 5

[24] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained k-means clustering with background knowledge. In C. E. Brodley and A. P. Danyluk, editors, *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 577–584. Morgan Kaufmann, 2001. 6

[25] X. Zhou, D. Wang, and P. Krähenbühl. Objects as points. *CoRR*, abs/1904.07850, 2019. URL http://arxiv.org/abs/1904.07850. 6

[26] S. Kornblith, M. Norouzi, H. Lee, and G. E. Hinton. Similarity of neural network representations revisited. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR, 2019. URL http://proceedings.mlr.press/v97/kornblith19a.html. 6