Supplementary Materials for MUSE-VAE

In this supplement, we provide additional details about the proposed MUSE-VAE, as well as the experimental evaluations, beyond those in the Main paper. Appendix A offers dataset specifications for SDD, nuScenes, and PFSD, with scene examples of each dataset. Appendix B elaborates on the implementation details, including the model networks and the approach we used to create the local view of the semantic map. In Appendix C, we define the evaluation metrics used in the Main paper. Appendix D presents details of two statistical significance tests, the Friedman test used in the Main paper, and the Bayesian Signed Rank test, whose results are shown here. Both tests offer additional evidence in support of improvements that MUSE-VAE framework makes beyond the baseline models. Appendix E supplement the qualitative analyses in the Main paper, showcasing instances of scenarios and the predictions made by all models in those scenarios, to highlight the different effects those models have on the forecasting process. Appendix F shows the limitation of the SDD segmentation provided by Y-net [29] to explain the low ECFL discussed in Sec. 4.2 of the Main paper. Finally, in Appendix G, we discuss some key challenges of the trajectory prediction model and suggest possible directions for future research.

A. Datasets

A.1. Real World Datasets

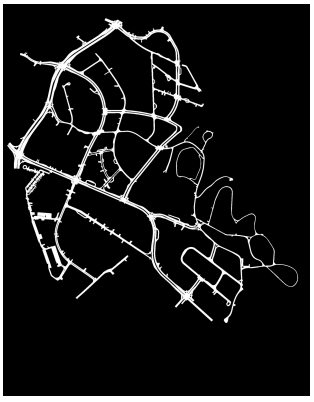
The **Stanford Drone Dataset** (SDD) [34] consists of 20 unique scenes of college campus from bird-eye view collected by drones. It contains various agents such as pedestrians, cyclists, skateboarder, cart, car, and bus. We use the same split following the TrajNet challenge [36]. As in [29, 37], we sample at 2.5 Hz, which yields 3.2s (8 frames) observed trajectories and 4.8s (12 frames) future trajectories. We take advantage of the semantic map as well as the pixel data processed by [29]. The semantic segmentation map is labeled as 5 classes; pavement, road, structure, terrain, and tree where each class has the class ID 1, 2, 3, 4, and 5, respectively. A sample scene image and its semantic map from SDD is shown in Fig. A.1a.

The **nuScenes Dataset** [6] is a public autonomous driving dataset. It provides 1,000 scenes in Boston, USA and Singapore and the corresponding HD semantic map with 11 annotated classes. Each scene is annotated at every 0.5s (2 Hz). Following the nuScenes prediction challenge setup, we split the train/val/test set, and predict only the vehicle category for 6s (12 frames) future trajectories based on 2s (4 frames) observations as in [28, 31, 50]. Fig. A.1b shows the global view of the binary map of the scene in Singapore with drivable (white-colored) area and undrivable (black-colored) area that nuScenes dataset provides.

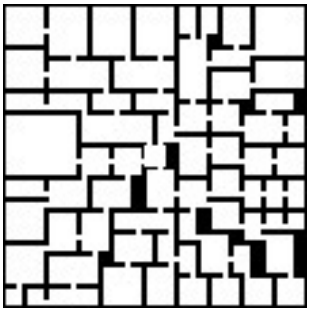
A.2. Synthetic Dataset

The **Path Finding Simulation Dataset** (PFSD) was generated by simulating the navigation of agents within 100 large synthetic environments borrowed from [43]. These environments were designed according to the external shapes and interior organizations of rooms and corridors generally found in contemporary architecture [10]. Unlike SDD and nuScenes, the non-navigable spaces in these environments are significantly more complex for navigation. Each of the environments was used to simulate 500 scenes (amounting to 50,000 total scenes), where a single agent navigates between two random points within the environment using the prevalent Social Force model [16]. As with SDD, the scenes were sampled at 2.5 Hz and further divided into training/val/test cases with 3.2s (8 frames) of observed trajectories and 4.8s (12 frames) of future trajectories. We use subset of the PFSD and make the train/val/test set with 40/2/4 different synthetic environments, respectively. We provide an environment example in Fig. A.1c. It is the bi-

(a) SDD



(b) nuScenes



(c) PFSD

Figure A.1. (a) The global view of the scene image (left) and the semantic map (right). The global view of the semantic map of (b) nuScenes, (c) PFSD.

nary map consisting of navigable (white-colored) and non-navigable (black-colored) space of the entire environment of one scenario. An agent finds a path by moving from a room to another room using the exit between obstacles.

B. Implementation Details

B.1. Local Semantic Map

Stanford Drone Dataset (SDD)
We divide the semantic map class values (1 through 5) from Y-net [29] with 5 so that the class values become 0.2 (pavement), 0.4 (road), 0.6 (structure), 0.8 (terrain), 1 (tree). We center the local view of the semantic map at the last observed step. As real-world agents have varying lengths of trajectories, for the radius of the local map we compute the per-step traversed distance of all trajectories, in each sequence (20 frames), and set the radius to be 20 times larger than the per-step distance. Because the local semantic map is centered at the last observed position, it is possible that the local map region exceeds the original map. We represent those areas not in the original map as 'non-navigable' space. We assume 'structure' is the most non-navigable space among the five aforementioned classes, thus pad those areas with 'structure' class value. Each of the local map images and the Gaussian heatmaps for trajectories is resized into 256x256 pixels, then concatenated in the channel dimension.
nuScenes Dataset
We use the official code of AgentFormer [50] to preprocess the nuScenes semantic map. This results in a 3 channel semantic map with four categories: drivable area, lane, road segmentation, and undrivable area. We further preprocess this information to create a single-channel semantic map by setting the drivable area, lane, road segmentation, and undrivable area as 0, 0.3, 0.6, and 1, respectively. To determine a local map size, we use the same policy as in SDD. For the local map region out of the original map, we pad it with the 'undrivable area' class value.
Path Finding Simulation Dataset (PFSD)
Since the synthetic dataset has consistent step size throughout the data, we compute the average per-step distance across the entire training set, about 8-pixel distance. Based on this, the local view of the semantic map is centered at the last observed position of the agent and its size is 160×160. We encode the navigable / non-navigable space as the values 0 / 1, respectively. The areas of the local map that deviate from the original map are padded by value 1 to indicate non-navigable space.

B.2. Networks

We implement `MUSE-VAE` in PyTorch. All networks are trained with Adam optimizer [20]. LG-CVAE has the backbone of U-net [35] combined with CVAE. U-net encoder blocks consist of [32, 32, 64, 64, 64] output channel dimensions with the input channel 2 consisting of a local map and a heatmap for past trajectories. The decoder blocks have [64, 64, 64, 32, 32] output channel dimensions with the final output channel 1 to predict the long-term goal heatmap. The posterior network consists of convolutional layers with same output channels as the U-net encoder blocks. The prior network takes the feature from the U-net encoder and process it further using two convolutional layers with output channel dimension [32, 32]. Following [22], the resulting 2D feature map is average-pooled into 1x1, then fed to a 1x1 convolutional layer to estimate the mean and the standard deviation of the posterior and the prior latent distribution, with the dimension set to 10. To avoid the posterior collapse, the encoder of LG-CVAE is pretrained with the AE loss for 10 epochs with the learning rate of $1e^{-3}$. During training of LG-CVAE with VAE loss, we anneal the KL loss for the first 10 epochs; FB is set as 0.7, 6, and 3 for PFSD, SDD, and nuScenes, respectively. The learning rate is $1e^{-3}, 1e^{-4}$, and $1e^{-4}$ for PFSD, SDD, and nuScenes, respectively.

SG-net is also based on the U-net. It has one additional block of 128 output channel dimensions more than LG-CVAE. The input channel of the encoder is 3, for a local map, a heatmap for past trajectories, and a heatmap for a long-term goal. The final output channel of the decoder is $N_{SG} + 1$ for N_{SG} heatmaps of N_{SG} short-term goals and a heatmap of a long-term goal. The learning rate is $1e^{-3}, 1e^{-4}$, and $1e^{-3}$ for PFSD, SDD, and nuScenes, respectively.

In Micro-net, we utilize the position, velocity, and acceleration of the past sequence as in [38]. The prior network consists of an LSTM with 64 hidden dimensions and 2 FC layers with the output dimensions [256, 40] to estimate the mean (20D) and the standard deviation (20D) of the prior latent distribution. The 256 dimensional hidden feature from the prior network is processed once more by concatenating it with the feature from the LG-CVAE, which encodes the semantic map using FC layer with 32 output dimensions in order to give the map information to the decoder. The posterior network consists of a bi-directional LSTM with 64 hidden dimensions, followed by two FC layers with [256, 40] to estimate the mean (20D) and the standard deviation (20D) of the posterior latent distribution. The decoder has a GRU with 128 hidden dimensions, followed by FC layers to predict the mean and standard deviation of the 2D position distribution. The short-term goal heatmap predictions from Macro-stage are converted to the 2D position and encoded by bi-directional LSTM with 64 hidden dimensions and further processed into a 2D feature by FC layer and fed to the GRU. We use the learning rate $1e^{-3}$ and $\beta = 50$ for all datasets. FB is 0.07 for PFSD and nuScenes, and 1 for SDD.

A subset of our code for PFSD is provided as additional supplementary material.The complete code for `MUSE-VAE` will be released upon acceptance, following the conference policy.

C. Evaluation metrics

To evaluate the performance, we use four metrics: Average Displacement Error (ADE), Final Displacement Error (FDE), Kernel Density Estimate-based Negative Log Likelihood (KDE NLL), and Environment Collision-Free Likelihood (ECFL).

Average Displacement Error (ADE) Given t_f future timestamps, ADE is defined as the L_2 distance between the future GT and predictions which is averaged over t_f. Following prior works [1, 15, 19, 29, 38, 50], we report the minimum ADE among K ADEs obtained from K predictions.

Final Displacement Error (FDE) FDE is L_2 distance between the GT and prediction at the final future step t_{p+f}. Same as ADE, the minimum FDE among K predictions is reported.

Kernel Density Estimate-based Negative Log Likelihood (KDE NLL) To determine if the generative model learns the characteristics such as variance and multi-modality of the distribution, [19, 38] introduce KDE NLL. First, the pdf is estimated by Kernel Density Estimate (KDE) using the K sampled predictions at each future timestep, and then the mean log-likelihood of the GT trajectory is obtained based on the pdf. We adopt the approach in [38] and their publicly released code.

Environment Collision-Free Likelihood (ECFL) Realistic trajectory predictions should not violate environmental restrictions. [42] proposes ECFL, the probability an agent has a path that is free of collision with the environment defined as $ECFL(p, E) = \frac{1}{k}\sum_{i=1}^{k}\prod_{t=1}^{t_f} E[p_{i,t,0}, p_{i,t,1}]$, where E is the scene environment represented as a binary map with 1s and 0s indicating the navigable and the non-navigable spaces, respectively. p are the K predicted positions of an agent under the temporal horizon t_f. We report ECFL in percent points, where 100% means no collisions.

D. Statistical Validity Test

Since our evaluation used multiple datasets and four measures, we conducted additional analyses using the average rank [9] and the Bayesian statistical validity analysis [2] to assess the significance of the obtained results.

D.1. The Friedman Test

We borrow notations from [9] in this section. We first calculated the Friedman statistic [12] as

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right], \tag{5}$$

where we compare k methods tested on N datasets. Here, R_j denotes the average ranks of algorithm j over all N datasets, i.e.,

$$R_j = \frac{1}{N}\sum_i r_i^j, \tag{6}$$

where r_i^j denotes the rank of j-th method among k algorithms tested on i-th dataset among N total datasets. One can approximate the probability distribution of the value as a Chi-square distribution. If k or N is small, one needs to find exact critical value from the precomputed table. Iman and Davenport [18] proposed a better statistic using the χ_F^2,

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2} \tag{7}$$

and this follows the F-distribution with $(k-1)$ and $(k-1)(N-1)$ degrees of freedom.

In our case, we have four methods to compare ($k = 4$) and 24 datasets ($N = 24$). We considered each evaluation setting (hyperparameter (K) choices, datasets, performance measures) as different datasets ($2 \times 3 \times 4$). Henceforth, we look for the F-distribution's critical value for 3 and 69 degrees of freedom. At 95% confidence level, the (upper) critical value is 2.737. Using the ranks obtained from our quantitative result, F_F is 21.278 which is significantly larger than the critical value, **rejecting the null hypothesis**, which states that all methods are equivalent.

As a post-hoc test, we conducted the Nemenyi test [30]. In the test, if two methods' average rank difference is larger than the critical difference defined as

$$CD = q_\alpha\sqrt{\frac{k(k+1)}{6N}}, \tag{8}$$

then there is a significant performance difference between the two methods. Here, q_α is 2.569 for $k = 4$ at 95% confidence level, hence $CD = 0.957$. Since the average rank of our method is 1.33 and that of AF is 2.33, we argue that **our method outperformed AF** in the evaluation. Note that the average ranks of Y-Net and T++ are 2.92 and 3.42, respectively.

D.2. Bayesian Signed Rank Test

We also provide significance testing result based on modern Bayesian statistical validity analysis to address potential limitations of the traditional frequentist null hypothesis significance testing [2]. We ran the Bayesian signed-rank test [3] for each pair of methods and for each measure. This test also accounts for the region of practical equivalence (ROPE) [24]. If the difference between two methods is smaller than the ROPE, then there is no practical difference in performance.

In our evaluation, we have several metrics and datasets and each needs a careful definition of ROPE to conduct a proper analysis. First, for ADE and FDE, we adopted the standard 0.5 meter difference as the ROPE [38]. However, one of the datasets we used, SDD, does not have the geometric calibration data to obtain metered measures in its test set, unlike PFSD and nuScenes. Henceforth, prior works, e.g., [29], used pixel differences to calculate the ADE and FDE. Therefore, we used 1 pixel difference for the ROPE, considering the resolution of the image and approximate sizes of real world structures and objects in the scene. It should be noted that, we also tested with a larger ROPE (3 pixels), but there was no change in the conclusion of this analysis. For KDE NLL, it is challenging to define ROPE since NLL is not a scale, but a likelihood value. So we set ROPE as zero for NLL. For ECFL, since it has same scale as accuracy $[0, 100]$, we use the standard 1% difference for the ROPE.

In Tabs. D.1 to D.4, we report the Bayesian signed-rank pairwise test result for $C(4, 2) = 6$ comparisons. Tab. D.5 summarizes all the aforementioned pairwise results by computing the average ranks of each method, in each of the tables Tabs. D.1 to D.4, based on the number of times the method "won", "tied", or "lost" in the pairwise comparison. For instance, in Tab. D.3, `MUSE-VAE` won 3-out-of-6 times, T++ 2/6, AF 1/6, and Y-Net 0/6 times, resulting in ranks of 1, 2, 3, and 4 for the four methods, respectively. Based on this, and in line with the traditional frequentist analysis in Sec. D.1, we conclude that **our `MUSE-VAE` outperforms the SOTA competitors**, on average, across all datasets and measures.

E. Additional Qualitative Evaluations

Fig. E.1 shows qualitative results in the same manner as those presented in Fig. 4 of the Main paper. Here we investigate several key scenarios from each dataset, beyond the'fork-in-the-road' introduced in Fig. 4. Scenarios were selected to highlight the challenges all models face in forecasting the environment-aware trajectories and offer insights into how the models behave when faced with environment constraints, in order to reveal the models' benefits and downsides.

The difference in the environment configurations between the two PFSD instances, Fig. E.1b here and Fig. 4b in the Main paper, is that Fig. E.1b has no obstacles in the direction of the observed, past trajectory while Fig. 4b presents obstacles at the bottom of the map in the same direction. Thus, comparing the predictions, our method predicted both straight ahead and left or right curved trajectories for Fig. E.1b, while producing only left or right curves for Fig. 4b.

Fig. E.1d shows an example when the ground truth trajectory passes right next to the 'structure' area, which is non-navigable; the heading direction is mostly blocked by the structure. Our model can make predictions that do not violate the environmental constraints, going back or turning left to search for navigable space. On the other hand, the predictions from the baseline models collide with the obstacles, a violation of the desired behavior.

Fig. E.1f is a fork-in-the-road scenario like Fig. 4f, with another drivable area on the other side of the fork in the

Table D.1. Comparing ADE of methods using Bayesian signed-rank test. For PFSD and nuScenes, ROPE is defined as 0.5 meters. For SDD, ROPE is 1 pixel.

PFSD, nuScenes				
Method A	p(A > B)	p(A ≈ B)	p(A < B)	Method B
T++	0.00	0.27	**0.73**	Y-Net
T++	0.00	0.27	**0.73**	AF
T++	0.00	0.27	**0.73**	Ours
Y-Net	0.00	0.27	**0.73**	AF
Y-Net	0.00	0.27	**0.73**	Ours
AF	0.00	**1.00**	0.00	Ours

SDD				
Method A	p(A > B)	p(A ≈ B)	p(A < B)	Method B
T++	0.00	**1.00**	0.00	Y-Net
T++	0.00	**1.00**	0.00	AF
T++	0.00	0.27	**0.73**	Ours
Y-Net	0.00	**1.00**	0.00	AF
Y-Net	0.00	**0.56**	0.44	Ours
AF	0.00	0.27	**0.73**	Ours

Table D.2. Comparing FDE of methods using Bayesian signed-rank test. For PFSD and nuScenes, ROPE is defined as 0.5 meters. For SDD, ROPE is 1 pixel.

PFSD, nuScenes				
Method A	p(A > B)	p(A ≈ B)	p(A < B)	Method B
T++	0.00	0.27	**0.73**	Y-Net
T++	0.00	0.16	**0.84**	AF
T++	0.00	0.20	**0.80**	Ours
Y-Net	0.00	0.27	**0.73**	AF
Y-Net	0.00	0.27	**0.73**	Ours
AF	0.00	**0.95**	0.05	Ours

SDD				
Method A	p(A > B)	p(A ≈ B)	p(A < B)	Method B
T++	0.00	0.04	**0.96**	Y-Net
T++	0.00	0.04	**0.96**	AF
T++	0.00	0.04	**0.96**	Ours
Y-Net	0.00	**1.00**	0.00	AF
Y-Net	0.00	**1.00**	0.00	Ours
AF	0.00	**1.00**	0.00	Ours

Table D.3. Comparing KDE NLL of methods using Bayesian signed-rank test. ROPE is 0 in this case.

Method A	p(A > B)	p(A ≈ B)	p(A < B)	Method B
T++	**1.00**	0.00	0.00	Y-Net
T++	**0.99**	0.00	0.01	AF
T++	0.00	0.00	**1.00**	Ours
Y-Net	0.31	0.00	**0.69**	AF
Y-Net	0.00	0.00	**1.00**	Ours
AF	0.00	0.00	**1.00**	Ours

Table D.4. Comparing ECFL of methods using Bayesian signed-rank test. ROPE is 1%.

Method A	p(A > B)	p(A ≈ B)	p(A < B)	Method B
T++	0.00	0.00	**1.00**	Y-Net
T++	0.00	0.01	**0.99**	AF
T++	0.00	0.00	**1.00**	Ours
Y-Net	0.03	0.27	**0.70**	AF
Y-Net	0.00	0.05	**0.95**	Ours
AF	0.00	0.03	**0.97**	Ours

Table D.5. Average rank of the four contrasted approached, based on the Bayesian Signed Rank pairwise test results in Tabs. D.1 to D.4, across all measures.

Method	Average Rank of Bayesian Test Results
T++	3.50
Y-net	3.00
AF	2.16
Ours	**1.33**

road. Although the traffic flow in this area is in the direction opposite to the predicted trajectory, it still is a drivable area. Since we have never provided a clear guidance for learning in which direction to drive based on the 'correct' lane, our model simply treats this area as drivable and makes one possible prediction. It can be seen that the baseline models cannot consider this possibility, instead making many predictions into the undrivable area.

Finally, Tab. E.1 shows the corresponding quantitative results for each dataset with metrics introduced in Appendix C. The results in the table are well-aligned with the visualization in the Tab. E.1. MUSE-VAE shows the highest ECFL in all datasets, suggesting our model forecasts environmentally-compliant trajectories. Moreover, our model shows the best performance for all datasets in terms of ADE; it similarly leads in FDE performance in SDD and nuScenes. MUSE-VAE attains the second best results in FDE for PFSD, trailing the top Y-Net by only 0.01 meter. Similarly, MUSE-VAE approaches the top method (AF) in KDE NLL for nuScenes. The third ranked performance of our model in KDE NLL of SDD stems from

Table E.1. Quantitative results of Fig. E.1. PFSD and SDD with $t_p = 3.2$s (8 frames) and $t_f = 4.8$s (12 frames), and nuScenes with $t_p = 2$s (4 frames) and $t_f = 6$s (12 frames). Errors are in meters for PFSD and nuScenes, and in pixels for SDD.

Dataset	Model	ADE ↓	FDE ↓	KDE NLL ↓	ECFL ↑
PFSD ($K = 20$)	T++	0.16	0.05	-1.54	95
	Y-net	0.1	**0.04**	-0.76	**100**
	AF	0.12	0.05	-0.50	**100**
	Ours	**0.08**	0.05	**-4.24**	**100**
SDD ($K = 20$)	T++	4.15	3.18	**6.58**	80
	Y-net	3.15	2.88	7.77	65
	AF	13.44	7.10	8.69	20
	Ours	**2.86**	**2.34**	8.45	**100**
nuScenes ($K = 10$)	T++	4.92	4.91	3.91	0
	Y-net	3.08	2.99	6.21	40
	AF	1.17	1.08	**3.68**	30
	Ours	**0.89**	**0.73**	3.84	**90**

those predictions heading to the left or going back toward the past trajectories. While away from the specific trajectory taken by the agent in this instance, the behaviors predicted by MUSE-VAE are very reasonable strategies for an agent who reaches a dead-end.

F. Limitation of SDD Segmentation

In our evaluations, we used the semantic map of SDD provided by Y-net. They classify the scene environment into the five classes described in Appendix B. In Fig. F.1, we show SDD scene images and their semantic maps of the scene (a) coupa_0 and (b) little_3. Red points indicate all trajectories in each scene.

There are two major problems in learning this map. First, it is not clear which semantic classes ought to be considered as navigable. Based on the class names, only pavement and road may be reasonably navigable space, but as seen in Fig. F.1, there are trajectories on tree, terrain, and structure. For the evaluation, we set only the 'structure' class as the obstacle class. Secondly, the segmentation regions are semantically inaccurate. Near the bottom-center of the semantic map of Fig. F.1a, we can see the squares all colored in yellow, which indicates a tree. However, looking at the scene image, we notice that not all of those regions are trees.

These inaccurate annotations give rise to the model confusion on how to deal with the map information when determining the trajectories that should only exist in navigable spaces. This affects MUSE-VAE more significantly than other models since the decision of long-term and short-term goals in Macro-stage heavily depend on the local map information; this subsequently leads to slightly lower ECFL for SDD in Sec. 4.2 of the Main paper, compared to other approaches.

G. Challenges and Future Work

In this paper, we proposed to "boost" the learning of models that forecast realistic, environment-aware trajectories by leveraging the large body of scene-compliant simulated trajectories in PFSD, a complex environment with intricate navigable / non-navigable structures. These structures were designed to induce diverse agent-environment behaviors, hence data to train models, that generalize well to many real-world scenarios.

Another component that makes trajectory prediction realistic is the absence of collisions among the agents themselves. One way to learn models that accomplish this, aside from collecting large bodies of real-world data, is to create synthetic datasets that reflect the desired agent-agent relationships, much like PFSD captures the environment-agent interactions. Such trained models would then transfer to the (smaller) real world datasets.

However, synthesizing collision-free models for agent-agent interactions is a challenging task. While designing scenarios where only the inter-agent distance is kept above a certain threshold is possible, such instances directly eliminate more complex yet desired behaviors such as agents walking together as a group or agents passing by each other in the opposite directions. Moreover, the behavioral patterns determining inter-agent proximity are also contextualized by the surrounding environment. For instance, the density of agents (hence their mutual displacements), will be higher (displacements lower) in very narrow navigable spaces compared to those in wider, open environments.

We leave it as an open research challenge to study such integrated models that can consider the inter-agent relationship in addition to the agent-environment interactions we tackled here using our `MUSE-VAE`.

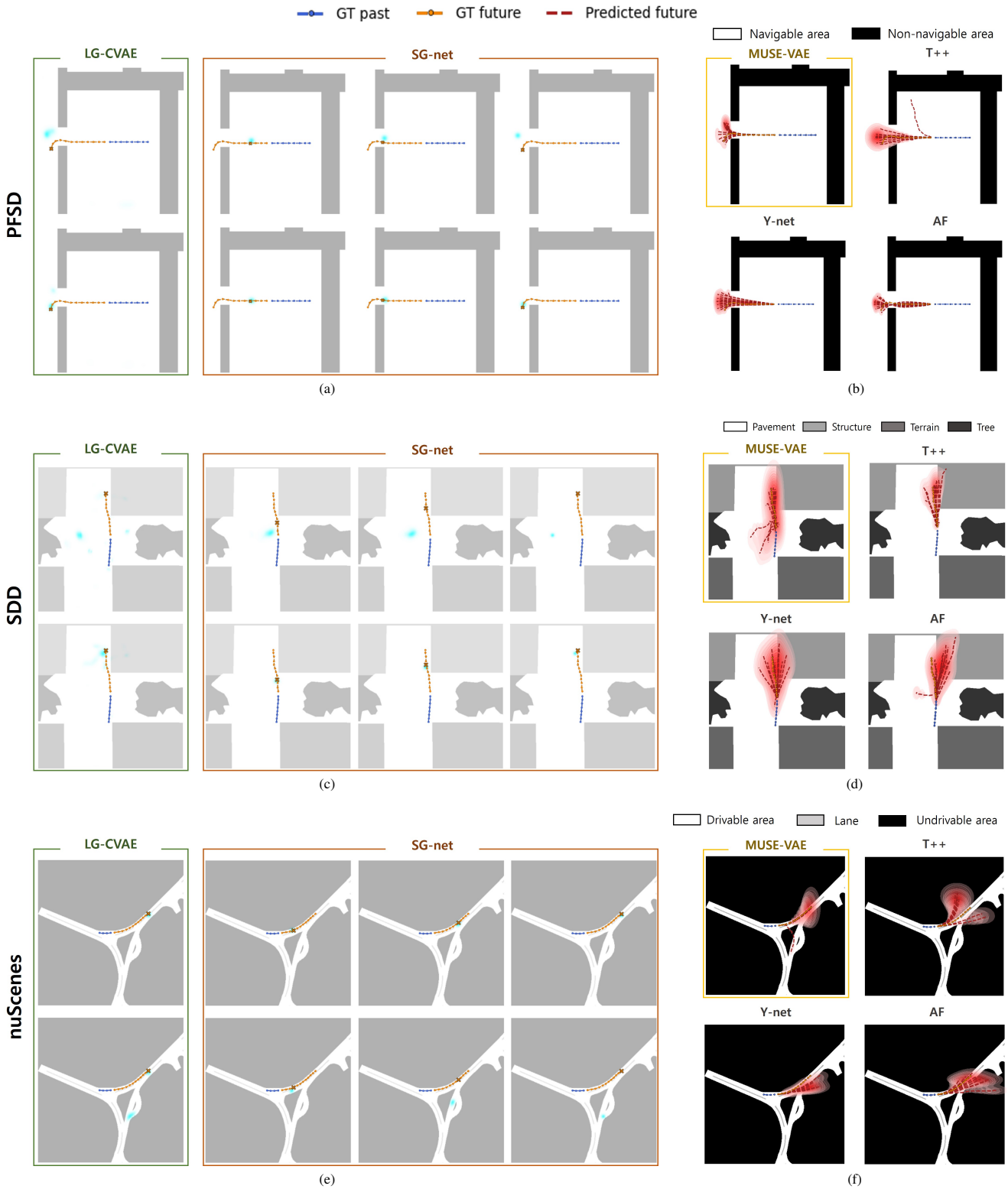
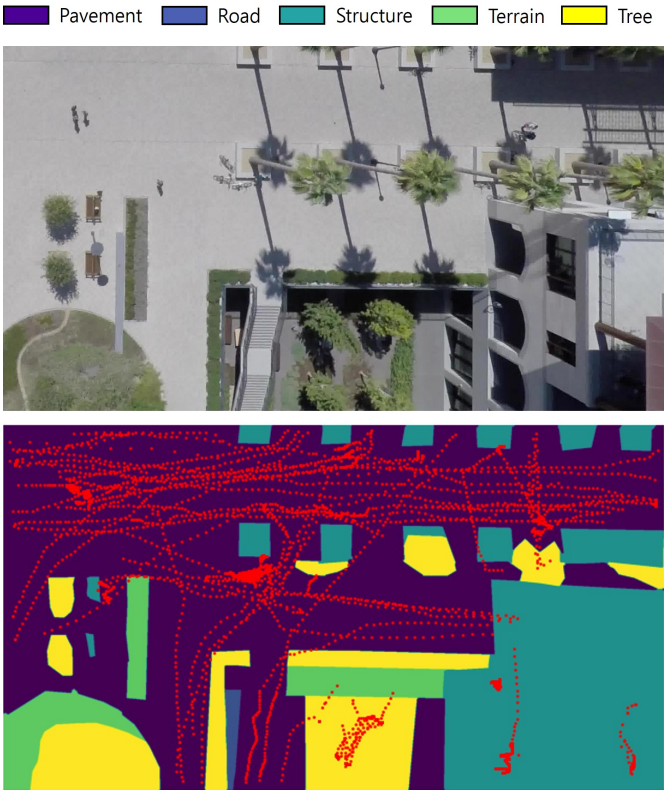


Figure E.1. Left: Macro-stage results of (a) PFSD, (c) SDD, and (e) nuScenes respectively. In the first column, the Long-term Goal (LG) heat map prediction from LG-CVAE is overlaid on the local semantic map. The following three columns are two Short-term Goals (SG) and one LG from SG-Net. Here we show only two different sampling generations in each dataset. The blue and orange lines indicate GT past and GT future trajectories, respectively. GT LG and SGs are marked with 'x'. Right: Complete trajectory predictions of (b) PFSD, (d) SDD, and (f) nuScenes respectively. In each dataset, the 1st/2nd/3rd/4th image from top-left to bottom-right is from Micro-stage of ours/Trajectron++/Y-net/AgentFormer, respectively. The blue, orange, and red lines indicate GT past, GT future, predicted future trajectories, respectively.



(a) coupa_0

(b) little_3

Figure F.1. SDD scene image and its semantic map of the scene (a) coupa_0 and (b) little_3. Red points indicate all trajectories in each scene. Trajectories are found in the region with classes like 'structure' or 'tree', which is unexpected in terms of navigability.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, 2016. 1, 2, 3

[2] Alessio Benavoli, Giorgio Corani, Janez Demšar, and Marco Zaffalon. Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis. *Journal of Machine Learning Research*, 18(77):1–36, 2017. 5, 3

[3] A. Benavoli, F. Mangili, G. Corani, M. Zaffalon, and F. Ruggeri. A bayesian wilcoxon signed-rank test based on the dirichlet process. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, page II–1026–II–1034. JMLR.org, 2014. 6, 3

[4] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *EMNLP*, 2015. 4

[5] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. Generating sentences from a continuous space. In *CoNLL*, 2016. 4

[6] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11618–11628, 2020. 1, 5

[7] Wang Yiwei Zhu Yiheng Cham Tat-Jen Cai Jianfei Yuan Junsong Liu Jun Cai, Yujun et al. A unified 3d human motion synthesis model via conditional variational auto-encoder. *Proceedings of the IEEE International Conference on Computer Vision*, 2021. 5

[8] Eric Chown, Stephen Kaplan, and David Kortenkamp. Prototypes, location, and associative networks (plan): Towards a unified theory of cognitive mapping. *Cognitive Science*, 19(1):1–51, 1995. 1

[9] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(1):1–30, 2006. 5, 3

[10] Timur Dogan, Emmanouil Saratsis, and Christoph Reinhart. The optimization potential of floor-plan typologies in early design energy modeling. 2015. 1

[11] Gonzalo Ferrer, Anais Garrell, and Alberto Sanfeliu. Social-aware robot navigation in urban environments. In *2013 European Conference on Mobile Robots*, pages 331–336. IEEE, 2013. 1

[12] Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701, 1937. 6, 3

[13] Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. Transformer networks for trajectory forecasting. *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10335–10342, 2021. 2

[14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 2

[15] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. pages 2255–2264, 06 2018. 1, 2, 3

[16] Dirk Helbing and Peter Molnar. Social Force Model for Pedestrian Dynamics. *Physical review E*, 51(5):4282, 1995. 1

[17] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017. 5

[18] Ronald L. Iman and James M. Davenport. Approximations of the critical region of the friedman statistic. *Communications in Statistics - Theory and Methods*, 9(6):571–595, 1980. 3

[19] B. Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2375–2384, 2019. 2, 5, 3

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 2

[21] Diederik P. Kingma, Tim Salimans, and Max Welling. Improved variational inference with inverse autoregressive flow. *ArXiv*, abs/1606.04934, 2017. 4

[22] Simon A. A. Kohl, Bernardino Romera-Paredes, Clemens Meyer, Jeffrey De Fauw, Joseph R. Ledsam, Klaus Maier-Hein, S. M. Ali Eslami, Danilo Jimenez Rezende, and Olaf Ronneberger. A probabilistic u-net for segmentation of ambiguous images. In *NeurIPS*, 2018. 4, 2

[23] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian D. Reid, Seyed Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In *NeurIPS*, 2019. 2

[24] John K. Kruschke and Torrin M. Liddell. The bayesian new statistics: Hypothesis testing, estimation, meta-analysis, and power analysis from a bayesian perspective. *Psychonomic Bulletin & Review*, 25(1):178–206, Feb 2018. 3

[25] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher Bongsoo Choy, Philip H. S. Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2165–2174, 2017. 2

[26] Bohan Li, Junxian He, Graham Neubig, Taylor Berg-Kirkpatrick, and Yiming Yang. A surprisingly effective fix for deep latent variable modeling of text. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Hong Kong, November 2019. 4

[27] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *2017*

IEEE International Conference on Computer Vision (ICCV), pages 2999–3007, 2017. 5

[28] Yecheng Jason Ma, Jeevana Priya Inala, Dinesh Jayaraman, and Osbert Bastani. Diverse sampling for normalizing flow based trajectory forecasting. *ArXiv*, abs/2011.15084, 2020. 5, 1

[29] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. In *Proc. International Conference on Computer Vision (ICCV)*, Oct. 2021. 1, 2, 3, 5, 4

[30] Peter B. Nemenyi. *Distribution-Free Multiple Comparisons*. PhD thesis, Princeton University, 1963. 3

[31] Tung Phan-Minh, Elena Corina Grigore, Freddy A. Boulton, Oscar Beijbom, and Eric M. Wolff. Covernet: Multimodal behavior prediction using trajectory sets. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14062–14071, 2020. 5, 1

[32] Bastiaan Quast. rnn: a recurrent neural network in r. *Working Papers*, 2016. 2, 3

[33] Nicholas Rhinehart, Rowan McAllister, Kris Kitani, and Sergey Levine. Precog: Prediction conditioned on goals in visual multi-agent settings. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2821–2830, 2019. 3

[34] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *ECCV*, 2016. 5, 1

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. volume 9351, pages 234–241, 10 2015. 4, 2

[36] Amir Sadeghian, Vineet Kosaraju, Agrim Gupta, Silvio Savarese, and A Alahi. Trajnet: Towards a benchmark for human trajectory prediction. *arXiv preprint*, 2018. 5, 1

[37] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. pages 1349–1358, 06 2019. 2, 5, 1

[38] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. *Trajectron++: Dynamically-Feasible Trajectory Forecasting with Heterogeneous Data*, pages 683–700. 12 2020. 2, 3, 5, 4

[39] Farnaz Sharif, Behnam Tayebi, György Buzsáki, Sebastien Royer, and Antonio Fernandez-Ruiz. Subcircuits of deep and superficial ca1 place cells support efficient spatial coding across heterogeneous environments. *Neuron*, 109(2):363–376, 2021. 1

[40] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning Structured Output Representation using Deep Conditional Generative Models. In *Neural Information Processing Systems (NIPS)*, 2015. 2, 4

[41] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *NIPS*, 2015. 5

[42] Samuel S. Sohn, Mihee Lee, Seonghyeon Moon, Gang Qiao, Muhammad Usman, Sejong Yoon, Vladimir Pavlovic, and Mubbasir Kapadia. A2x: An agent and environment interaction benchmark for multimodal human trajectory prediction. In *Motion, Interaction and Games*, MIG '21, New York, NY, USA, 2021. Association for Computing Machinery. 5, 3

[43] Samuel S Sohn, Honglu Zhou, Seonghyeon Moon, Sejong Yoon, Vladimir Pavlovic, and Mubbasir Kapadia. Laying the foundations of deep long-term crowd flow prediction. In *European Conference on Computer Vision*, pages 711–728. Springer, 2020. 5, 1

[44] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In *NIPS*, 2016. 4

[45] Yichuan Tang and Ruslan Salakhutdinov. Multiple futures prediction. In *NeurIPS*, 2019. 2

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. 2

[47] Anirudh Vemula, Katharina Muelling, and Jean Oh. Social attention: Modeling attention in human crowds. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7, 2018. 1, 2

[48] Chuhua Wang, Yuchen Wang, Mingze Xu, and David J. Crandall. Stepwise goal-driven networks for trajectory prediction. *ArXiv*, abs/2103.14107, 2021. 1, 3

[49] Jan M Wiener, Simon J Büchner, and Christoph Hölscher. Taxonomy of human wayfinding tasks: A knowledge-based approach. *Spatial Cognition & Computation*, 9(2):152–165, 2009. 1

[50] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 3, 5, 1

[51] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Benjamin Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, Congcong Li, and Dragomir Anguelov. TNT: target-driven trajectory prediction. *CoRR*, abs/2008.08294, 2020. 1, 2, 3