From Goals, Waypoints & Paths To Multimodal Human Trajectory Forecasting - Supplementary Material -

Karttikeya Mangalam[†*] Yang An[§*] Harshayu Girase[†] Jitendra Malik[†]

[†] UC Berkeley [§] Technical University of Munich

Figure 1: **Conditioned Waypoint sampling**: 1) The first image shows the scene with the past trajectory as blue circles. The yellow star indicates the sampled goal. 2) The unconditioned waypoint distribution can be seen in the second image. The distribution is goal-agnostic and therefore probability mass is distributed over all three roads. 3) The third image is the resulting waypoint distribution, by multiplying the multivariate Gaussian prior with the unconditioned prediction. The diamond indicates a sampled waypoint 4) The final image shows the trajectory towards the sampled goal (star) while crossing the sampled waypoint (diamond).

	SDD			inD	
TTST	✗	✗	✓	✗	✓
CWS	✗	✓	✓	✓	✓
ADE	65.00	52.31	47.94	17.77	14.99
FDE	86.98	86.98	66.71	28.52	21.13

Table 1: **Ablation results for Conditioned Waypoint Sampling (CWS) and TTST**: We benchmark the performance of Y-net with and without our proposed CWS and TTST on our long horizon forecasting setting, predicting $t_f = 30$ seconds into the future given $t_p = 5$ seconds of past motion history. All reported errors are in pixels (lower is better) for $N^w = 1$, $K_e = 20$ and $K_a = 1$.

1. Supplementary details for Y-net

1.1. Conditioned Waypoint sampling

Goal and waypoints are dependent to each other. Figure 1 shows an example, where the road forks into three different paths. If the sampled goal lies on the bottom path, the agent most likely will not pass through a waypoint on the upper or middle path, and will prefer a waypoint lying on the bottom road.

Following this intuition, we introduce a prior to condition the waypoint sampling hierarchically on the already sampled goal and waypoints.

We first sample and fix the goal and thereby the possible locations of the next waypoint is constrained. We assume that the waypoint lies on a straight line segment connecting

$*$ indicates equal contribution.

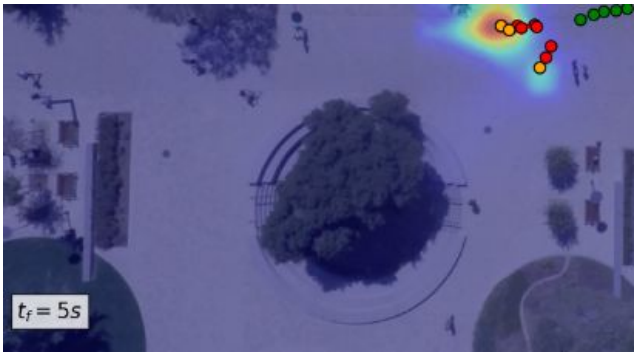

Figure 2: **GIF Visualization**: Demonstrating the goal, waypoint and path multimodality for long term human trajectory prediction (30 seconds). Given the past 5 seconds input history (green), we predict diverse future trajectories (current location in orange, past in red). *Due to restrictions, we can only show a snapshot. Please refer to the supplementary file or ArXiv version for the animation.*

the sampled goal and the past trajectory at exactly $\frac{w_i - n_p}{n_f}$ of the line segment length, where w_i denotes the chosen timestep for waypoint i, n_p the past timesteps and n_f is the future prediction horizon in timesteps, *e.g.* with the setting $N^w = 1$ waypoint at $w_1 = 20$, $n_p = 5$ and $N_f = 30$, it would lie in the middle of the segment. However, this assumption is too restrictive and it can lead to unrealistic paths not complying with the environment constraints. To relax this assumption, we use a multivariate Gaussian prior centered at the assumed location. The variance is chosen adaptively by considering the distance between the agent's current position and the sampled goal as $\sigma_\perp = \frac{||u_{n_p+n_f} - u_{n_p}||}{\alpha}$ where α is a scaling hyper parameter. We set $\alpha = 6$ in our experiments. Intuitively, the greater the distance between the current position and the sampled goal, the more uncertain is the waypoint position. $\sigma_\perp$ is the variance perpendicular to the line segment, and we set the variance parallel to the line segment to $\sigma_\parallel = \beta * \sigma_\perp$ with $\beta = 0.5$ in our experiments. This constrains the possible waypoint position more in the direction of travel and leaves more room for uncertainty in the perpendicular direction.

We multiply the described multivariate Gaussian prior to the predicted waypoint distribution and normalize the values. Fusing prior and predicted distribution leads to scene-compliant waypoints (predicted distribution) in the direction towards the sampled goal (prior). As seen in the example, it suppresses probability mass on the upper and middle road. From the resulting distribution we use `softargmax` for the first waypoint and sample the remaining $K_a - 1$ waypoints randomly according to the probability values of each pixel, as described in Section 3.2 in our main paper, to get two-dimensional points from the distribution.

If there is more than one waypoint, i.e. $N^w > 1$, we repeat the above process for the next waypoint at w_i and condition it to the previously sampled waypoint at w_{i+1}.

Table 1 shows an ablation of the Conditioned Waypoint Sampling `CWS`. While it doesn't have an effect on the FDE – `CWS` doesn't affect the goal sampling – the ADE decreases by 24.3%.

1.2. Test-Time Sampling Trick

In situations where multiple samples are required, such as during testing, sampling from $\mathbb{P}$ can be carried in a straightforward fashion by considering X as a categorical distribution with given probabilities X_{ij} for position (i, j). However, this approach doesn't take into account the number of samples required from $\mathbb{P}$ all of which jointly will represent the quality of the estimated $\mathbb{P}$. For example, if only a few samples are required, sampling indiscriminately spatially is sub-optimal since samples are likely to be wasted if drawn from low probability regions or sampled from neighboring regions.

Hence, we propose a 'Test-Time Sampling Trick' (`TTST`) that is cognizant of the number of goal samples, K_e, needed for evaluation. During testing, we propose to first sample a large number of points (10,000 in our experiments) from the estimated distribution $\hat{P}$ in a K_e-agnostic manner.

To eliminate outliers, we suppress samples from pixels (i, j) with probability X_{ij} below a threshold thr_{rel}. It is set adaptively for each probability matrix X separately to a fraction of the highest occurring probability in that matrix: $thr_{rel} = \max(X) * 0.01$.

Further, to control the tradeoff between diversity and precision, we use the temperature T. Before the pixel-wise sigmoid operation, we divide the predicted logit probability map through T. Lower temperature values results in probability maps X with low entropy, i.e. the probability mass is concentrated in a smaller number of pixels and samples are increasingly drawn from more likely regions. Higher temperature values increase the diversity of samples. For the short term setting, we use $T = 1.0$, while for our proposed long term setting, we increase the diversity by setting $T = 1.8$.

Then, we propose to run the fast clustering algorithm K-means on these sampled points with the number of clusters set to $K_e - 1$. The cluster centers obtained from the K-means algorithm, along with the softargmax sampled point, form the final set of K_e samples to be used for evaluation. Note that while in spirit this is similar to the 'truncation trick' proposed in PECNet [9], the 'truncation trick' is K-agnostic and requires a well suited σ_T to be chosen experimentally beforehand for a given K. Further, their 'truncation trick' operates in the latent variable space with no direct control over the final generated samples since in [9] multi-modality is introduced through implicit approaches like Variational Auto-encoders. Alleviating these limitations, `TTST` provides direct control on the sampled points,

is cognizant of the number of samples needed K and does not require any K-specific tuning because of the design choice of using explicit probability heatmaps.

Table 1 shows the effectiveness of our proposed TTST. TTST reduces the error on SDD and inD by 9.1% and 18.5% in ADE, respectively, and 30.4% and 35.0% in FDE.

2. Data Preprocessing

In this section we describe the data preprocessing steps for our proposed long term setting for both datasets: Stanford Drone Dataset (SDD) [12] and Intersection Drone Dataset (inD) [2].

For the short term setting on SDD and ETH [11] / UCY [7], we use the well-established preprocessed data from the TrajNet benchmark [14] and Social GAN [5] respectively. While SDD already lies in pixel coordinates, we describe our conversion process from world coordinates to pixel coordinates for ETH/UCY in Section 2.3

2.1. Stanford Drone Dataset

As mentioned above, we use the preprocessed data from the TrajNet benchmark [14] for the short term benchmark to be comparable with other state-of-the-art methods and their results.

For the proposed long term setting,we split the data of Stanford Drone Dataset (SDD) in the same fashion as proposed in TrajNet benchmark [14] evaluating on the same scenes, all of which are not seen during training. The raw data is recorded in $\texttt{FPS} = 30$ and we first downsample the data to our proposed $\texttt{FPS} = 1$, so one time step in the trajectory equals 1 second. The raw data contains bounding boxes of the detected agents. We use the middle point of the bounding boxes to get the same coordinate representation as the short term setting. The data contains various types of agents beyond pedestrians (bicyclists, skateboarders, cars, buses, and golf carts), we filter out all non-pedestrians and short trajectories below $n_p + n_f$ out. As the raw data is noisy and contains temporal discontinuities, we split the trajectories at those discontinuities. We use a sliding window approach without overlap to split up long trajectories, resulting in our final dataset.

2.2. Intersection Drone Dataset

We use similar steps for the Intersection Drone Dataset as for SDD Section 2.1. To evaluate Y-net and the baselines performance on unseen scenes during training, we only use location ID 4 during testing. The video and detection are in $\texttt{FPS} = 25$, and again, we downsample the data to $\texttt{FPS} = 1$. We then filter out non-pedestrians and short trajectories and use a sliding window approach without overlap to split long trajectories. Since the data lies in world coordinates, we convert it into pixel coordinates by scaling with the provided scale factors from the authors.

2.3. ETH/UCY

We use the same preprocessed data as [5] [1]. Our model represents trajectories as heatmaps and hence needs the coordinates in pixel space. The ETH/UCY data lie in world coordinates (meter unit). To project the data into pixel space we use the provided homography matrices from the dataset for the ETH [11] scenes ETH and HOTEL and create our own homography matrices for the UCY[7] scenes UNIV, ZARA1 and ZARA2. To enable fair comparisons, we convert our predictions back to world coordinates using the inverse homography matrices and calculate our errors with the untouched raw data in world coordinates, to avoid any errors from our projection.

3. Baseline models

We benchmarks against several state-of-the-art methods across both short and long term trajectory forecasting settings which we describe briefly.

- Social GAN [5] proposes a GAN for multi-modal trajectory forecasting lumping together the where/how multi-modalities.
- Conditional Flow VAE (CF-VAE) [1] uses a conditional normalizing flow based Variational auto-encoder that models future uncertainty without disentangling underlying factors.
- P2TIRL [4] is a grid based trajectory forecasting method learnt using maximum entropy inverse reinforcement learning.
- SimAug [8] is a recently proposed method that uses additional adversarially generated 3D multi-view data for adapating to novel viewpoints in forecasting.
- PECNet [9] is the prior state-of-the art method at the time of submission on short-term trajectory prediction on the Stanford Drone Dataset. They propose to use goal-conditioning but does not account for multi-modality in the path to the goal.
- DESIRE [6] proposes an inverse reinforcement learning approach for prediction by planning and uses a refinement module to rank and optimize the predictions.
- TNT [19] closely improves upon PECNet's performance for $K = 5$ samples on SDD and is the prior state-of-the-art in that setting.
- Trajectron++ [15] proposes a recurrent graph based forecasting model incorporating dynamic constrains such as other moving agents and scene information.

[1] https://github.com/agrimgupta92/sgan

- LB-EBM [10] is a recently proposed method that learns an energy-based model in the latent space and a policy generator to map the latent vector into a trajectory. It improved upon PECNet's performance and was the previous state-of-the-art method on SDD in the short term setting.
- Introvert [16] uses a 3D visual attention mechanism conditioned on the observed trajectory to extract scene and social information from videos.
- AgentFormer [18] is an attention based method that jointly models the time dimension and social interactions using a sequence representation while preserving each agents identity. This work also hold the prior state-of-the-art on ETH/UCY short term trajectory prediction benchmark.

4. Implementation Details

4.1. Segmentation Model

To incorporate constraints and interactions of the agents with the scene, we pretrain a semantic segmentation model to efficiently use the sparse scene image data. Stanford Drone Dataset contains 60 scene images in total, while inD only contains images from four different recording locations. We use the U-net model [13] with ResNet101 [**?**] backbone. The ResNet101 encoder's weights are pretrained on ImageNet, while the weights for the U-net decoder and segmentation head are randomly initialized. The images are downsampled by a factor of four (SDD) and three (inD), padded to be divisible by 32 as required for U-net and cropped to 256×256. The data is augmented spatially by rotation, flipping, scaling and perspective transformation and we introduce Gaussian noise, blurring as well as color, brightness and contrast shifts. The semantic maps are manually labeled into the $N_c = 5$ classes: Pavement, tree, terrain, structure and road, as well as a dummy class for padding and black areas in the inD dataset. We will release the labeled semantic segmentation maps for reproducibility and future work. We only use the corresponding images from the trajectory train scenes for training to evaluate the performance of Y-net on unseen environments for both SDD and inD.

The SDD segmentation model is trained using ADAM optimizer to reduce the Dice Loss [17] with an initial learning rate of 1×10^{-4} and batch size of 4. The learning rate is decreased to 1×10^{-5} after 1500 epochs. Further we freeze the ResNet101 backbone for the first 200 epochs.

As inD only contains images from four locations, we use the pretrained SDD model and freeze the encoder weights for the first 1000 epochs to avoid catastrophic forgetting. All other hyper parameters are the same as for training SDD.

4.2. Training

We train the entire network end to end with ADAM optimizer with a learning rate of 1×10^{-4} and batch size of 8. We scale the overall loss by a factor of 1000. Since the scene images $\mathcal{I}$ have different heights and widths in all datasets, we ensure that each batch only contains the image and trajectories from the same scene. Y-net does not use fully-connected layers and therefore can handle images of different sizes, without cropping or padding to the same shape. The RGB scene image $\mathcal{I}$ and trajectory heatmaps $\mathbf{H}$ are downsampled by 4 for SDD, 3 for inD and 1.5 for ETH/UCY to save memory and padded to be divisible by 32. For fair comparisons with previous methods we upsample the predicted trajectories back to its original size and compare with the ground-truth data in original scale. All scene images and trajectories are augmented by spatial flipping and rotation in 90° steps, effectively increasing the number of training trajectories by a factor of 8. The encoder blocks in $\mathbf{U}_e$ have output channel dimensions $[32, 32, 64, 64, 64]$ and both $\mathbf{U}_g$ and $\mathbf{U}_t$ start with two convolutional layers of output channel dimensions 128, followed by blocks of output channel dimensions $[64, 64, 64, 32, 32]$. We use $\lambda_1 = \lambda_2 = 1$ to weight the binary cross entropy loss between $\mathbf{U}_g$ and $\mathbf{U}_t$.

During training $\mathbf{U}_g$ predicts the goal and waypoint distribution for all n_p time steps as an auxiliary task. We presume that this helps to let the sub-network learn the dynamics of pedestrian trajectories better. During inference, we only use the goal and N^w waypoint distributions as needed.

The trajectory sub-network $\mathbf{U}_t$ is trained using the ground-truth goal and waypoints. Those are represented as trajectory heatmaps as described in Section 3.1 in our main paper and downsampled spatially to fit the corresponding feature map shapes of $\mathbf{U}_t$ blocks. By using the ground-truth, $\mathbf{U}_t$ learns to predict trajectories leading towards the goal, while passing the waypoints. During inference, we use the (`TTST`) sampled goals and waypoints predicted by $\mathbf{U}_g$.

We further experiment with deformable convolutional layers as proposed in [3] on ETH/UCY.

5. Additional Qualitative Results

We show additional qualitative results for long term trajectory prediction ($t_f = 30$) on SDD through a GIF temporally on another scene in Figure 2.

References

[1] Apratim Bhattacharyya, Michael Hanselmann, Mario Fritz, Bernt Schiele, and Christoph-Nikolas Straehle. Conditional flow variational autoencoders for structured sequence prediction. *arXiv preprint arXiv:1908.09008*, 2019.

[2] Julian Bock, Robert Krajewski, Tobias Moers, Steffen Runde, Lennart Vater, and Lutz Eckstein. The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections. 2019.

[3] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[4] Nachiket Deo and Mohan M Trivedi. Trajectory forecasts in unknown environments conditioned on grid-based plans. *arXiv preprint arXiv:2001.00735*, 2020.

[5] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018.

[6] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 336–345, 2017.

[7] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer graphics forum*, volume 26, pages 655–664. Wiley Online Library, 2007.

[8] Junwei Liang, Lu Jiang, and Alexander Hauptmann. Simaug: Learning robust representations from 3d simulation for pedestrian trajectory prediction in unseen cameras. 2020.

[9] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. *arXiv preprint arXiv:2004.02025*, 2020.

[10] Bo Pang, Tianyang Zhao, Xu Xie, and Ying Nian Wu. Trajectory prediction with latent belief energy-based model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11814–11824, 2021.

[11] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pages 261–268. IEEE, 2009.

[12] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European conference on computer vision*, pages 549–565. Springer, 2016.

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

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

[15] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. *arXiv preprint arXiv:2001.03093*, 2020.

[16] Nasim Shafiee, Taskin Padir, and Ehsan Elhamifar. Introvert: Human trajectory prediction via conditional 3d attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16815–16825, 2021.

[17] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 240–248. Springer, 2017.

[18] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. *arXiv preprint arXiv:2103.14023*, 2021.

[19] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Benjamin Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, et al. Tnt: Target-driven trajectory prediction. *arXiv preprint arXiv:2008.08294*, 2020.