Supplementary Material: Adaptive Human Trajectory Prediction via Latent Corridors

Neerja Thakkar[1], Karttikeya Mangalam[1], Andrea Bajcsy[2], and Jitendra Malik[1]

[1] UC Berkeley
[2] Carnegie Mellon University

In the supplementary material, we provide an ablation on the implementation of our prompting method and a visual overview of our ATP problem formulation. We also show additional qualitative results on MOTSynth and more detailed quantitative results on EarthCam data, quantitative results on ETH/UCY, and compare our implementation of the scene-aware baseline to the original YNet paper [1]. Video results can be seen at this webpage.

1 Details on Latent Corridors for Adaptive Trajectory Prediction

1.1 Adaptive Trajectory Prediction Visualization

We provide an illustrative overview of adaptive trajectory prediction problem, formulated in main text Sec. 3, in Fig. 1.

1.2 ATP via Latent Corridors on Architectures Beyond YNet

ATP is an architecture-agnostic paradigm, and latent corridors are also not specific to Y-Net but rather can also work on different architectures. To demonstrate this, we experimented with the Learned Trajectory (PECNet-Ours) architecture on the 473 MOTSynth scenes. Taking the pretrained PECNet-Ours model, we summed a per-scene 16 parameter latent directly to the input, eight xy coordinates. Training the 16D latent along with finetuning the last layer of PECNet-Ours, we see an improvement of 10.2% on ADE and 10.4% on FDE.

1.3 Ablation on Prompt Location/Size

We ablated the prompt location and method of combining with the input. For the input location, we experimented with combining the prompt with several parts of the input to $\mathcal{P}$, $[\mathbf{M}_{\tau-H:\tau}, S]$: all of the input heatmaps $\mathbf{M}_{\tau-H:\tau}$, just the first input heatmap $\mathbf{M}_0$, just the segmentation map S, and all of the inputs. For method of combination, we experimented with element-wise summing and element-wise multiplication. We ran this ablation on the latent corridors-only approach to training on MOTSynth. Results can be seen in Table 2. While summing seems to lead to better performance than multiplying, and summing to the heatmaps seems to be more helpful than summing to the segmentation,

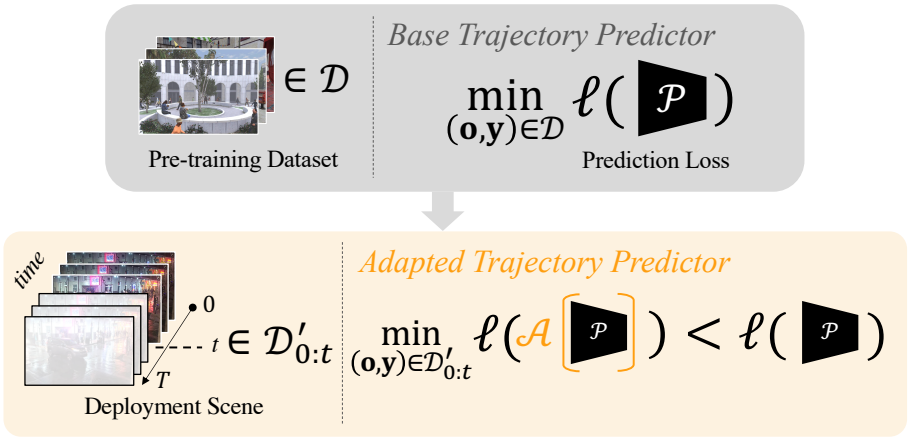

Fig. 1: Adaptive trajectory prediction. ATP, formulated in Sec. 3, allows a pre-trained predictor $\mathcal{P}$ to adapt to a new deployment scene by learning over time on the deployment scene. Once adaptation has occurred, the adapted predictor $\mathcal{A}[\mathcal{P}]$ should perform better on the deployment scene.

Method	ADE	FDE
Learned Trajectory (PECNet-Ours)	51.2	100.0
ATP (LC + Joint Finetune)	**46.0**	**89.6**

Table 1: Results of applying ATP via latent corridors to PECNet. ATP using 16D latent corridors with joint finetuning improves the performance of PECNet.

generally, the prompts are effective at improving performance on a variety of input locations. This shows promise for training latent corridors to adapt a variety of architectures.

1.4 Segmentation Classes

We condense Mask2Former's 132 classes into 12 classes that are meaningful for outdoor pedestrians: person, bicycle, car, motorcycle, large vehicle, traffic light, stop sign, bench, stairs, road/ground, building/wall, other.

2 Additional Experimental Results

2.1 Results on MOTSynth, WildTrack, MOT, and EarthCam

In Fig. 2, we see the FDE results for MOTSynth, WildTrack and MOT over time. Similar to results with the ADE metric, we see that on real data (Fig. 2a), the adaptive finetuning baseline is slightly better than just latent corridors, but ATP via both latent corridor prompting and per-scene finetuning largely outperforms both of those, and all adaptive methods outperform the non-adaptive baselines.

Prompt Method	*ADE*	*FDE*
Sum to all heatmaps $\mathbf{M}_{\tau-H:\tau}$	**44.6**	**90.2**
Sum to $\mathbf{M}_0$	45.1	92.1
Sum to S	46.4	97.3
Sum to $\mathbf{M}_{\tau-H:\tau}$ and S	46.4	96.5
Multiply to all heatmaps $\mathbf{M}_{\tau-H:\tau}$	46.5	96.1
Multiply to S	45.8	93.6
Multiply to $\mathbf{M}_0$	47.6	101.6
Multiply to $\mathbf{M}_{\tau-H:\tau}$ and S	46.5	96.1

Table 2: Ablation on 437 MOTSynth scenes in the ATP latent corridor adaptation configuration. We experiment with different locations and methods of combining the prompt with the input, and find that summing the prompt to all input heatmaps yields the best result, but most combinations result in an improvement on the scene-aware baseline (see main text Table 1).

On the MOTSynth data, as with ADE, the ATP via just per-scene finetuning or just latent corridors are comparable (Fig. 2b). With FDE, while some scenes have minimal gains, we see even more significant error reduction for some scenes than with ADE, of up to 91.4%, and an average FDE improvement of 33.9% from ATP via latent corridors and per-scene finetuning on 25 MOTSynth scenes (Fig. 2c).

We showcase additional qualitative results on MOTSynth data in Fig. 3. We see that our approach is able to learn that in a nighttime scene with a large staircase, pedestrians mostly move towards the staircase, regardless of the direction of their observed history (top left and middle). We also see that our approach learns that pedestrians tend to stay on a walkway (top right). Finally, we see several examples of our approach having awareness of the 3D ground plane and predicting future trajectories that lie within the ground plane (bottom two rows).

Quantitative results for each of the EarthCam scenes can be seen in Table 3. We see that across the four EarthCam scenes, ATP via per-scene finetuning alone works better than using latent corridor adaptation alone (by a narrow margin on both NoLA scenes, and by a significant amount on the Rick's Cafe scene), but a combination of the two is significantly better than any other adaptive or non-adaptive approach. Interestingly, for the Rick's Cafe and Times Square scenes, a constant velocity baseline is better than our non-adaptive learned baselines, but our ATP approach outperforms all baselines.

2.2 Results on ETH/UCY

The main text focused on challenging datasets with non-top-down camera viewpoints, as compared to birds-eye-view datasets popular in earlier works such as SDD and ETH/UCY. Here, we run our approach in this setting by evaluating on ETH/UCY. For each scene in ETH/UCY (ETH, HOTEL, UNIV, ZARA1, ZARA2), we construct an 80/20 train-test split and evaluate as described in main

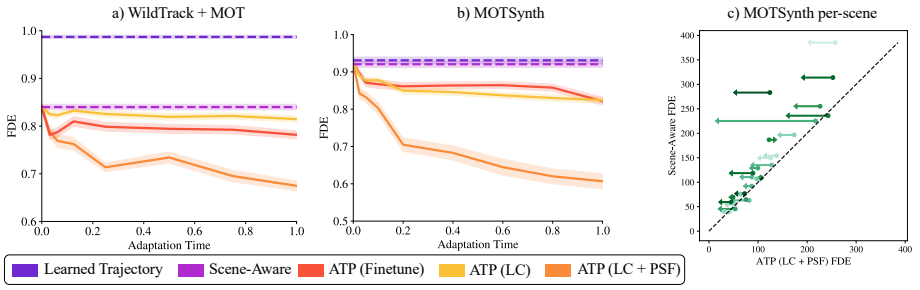

Fig. 2: Adaptation over time FDE. As in main text Fig. 5, the x-axis represents normalized adaptation time in human-seconds. The y-axis represents the FDE (lower is better). Results are normalized per-scene and averaged over models trained on 25 MOTSynth scenes (a) and 7 from MOT and WildTrack (b), with shaded area $\sigma/10$. For the FDE metric, our methods improve on the baselines increasingly with adaptation time. Latent corridors + per-scene finetuning has the best performance, as with FDE, and ATP via just finetuning or just latent corridor learning is still comparable. c) Comparison to baseline over many MOTSynth scenes for models trained with 8% (point) and 80% (arrowhead) human-second datasets for FDE. For many deployment scenes, FDE improves significantly more with our method than ADE improved, but still, the per-scene improvements are varied.

text section 6.2. Results in Table 4 are in pixels and we compute ADE/FDE on a single predicted future. We see a 4.6% improvement in ADE and 2.4% in FDE using LC over per-scene finetuning.

3 Choice of Baselines

Our key scene-aware baseline is YNet, which outperforms or is comparable to other methods that utilize scene priors and trajectory histories such as [2–4, 6]. Since we used a simplified version of the YNet architecture for our scene-aware baseline, we have further benchmarked against the original YNet using the codebase training configuration using MOTSynth and the Stanford Drone Dataset [5]. We see in Table 5 that the YNet-Ours outperforms the original YNet in the unimodal setting.

References

1. Mangalam, K., An, Y., Girase, H., Malik, J.: From goals, waypoints & paths to long term human trajectory forecasting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15233–15242 (2021)
2. Manh, H., Alaghband, G.: Scene-lstm: A model for human trajectory prediction. arXiv preprint arXiv:1808.04018 (2018)
3. Marchetti, F., Becattini, F., Seidenari, L., Bimbo, A.D.: Mantra: Memory augmented networks for multiple trajectory prediction. In: Proceedings of the

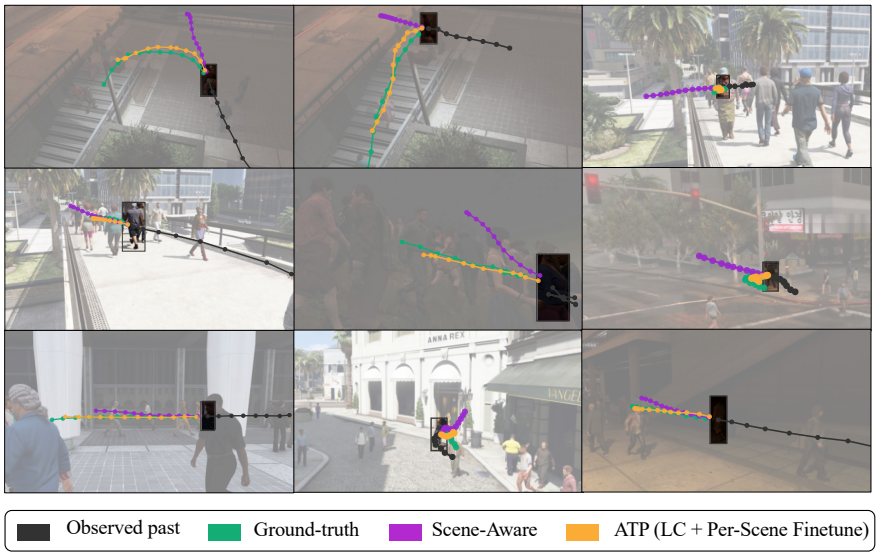

Fig. 3: Additional MOTSynth qualitative results. (top left and middle) From several examples of pedestrians in motion, our approach (orange) is able to learn that in this scene, most pedestrians will turn to go down the stairs, while the scene-aware baseline (purple) struggles to understand this scene-specific feature, and instead assumes that the pedestrian will continue walking in the direction of the observed history. Our model is also able to gain understanding that most pedestrians will stay on a walkway, even if they move in a direction orthogonal to it (top right). We also see many more examples of our approach having better awareness of the 3D nature of the ground plane projected into the 2D image (bottom two rows), even when the ground plane is tilted (middle middle).

IEEE/CVF conference on computer vision and pattern recognition. pp. 7143–7152 (2020)
4. Meng, M., Wu, Z., Chen, T., Cai, X., Zhou, X., Yang, F., Shen, D.: Forecasting human trajectory from scene history. Advances in Neural Information Processing Systems **35**, 24920–24933 (2022)
5. Robicquet, A., Sadeghian, A., Alahi, A., Savarese, S.: Learning social etiquette: Human trajectory understanding in crowded scenes. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14. pp. 549–565. Springer (2016)
6. Xue, H., Huynh, D.Q., Reynolds, M.: Ss-lstm: A hierarchical lstm model for pedestrian trajectory prediction. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1186–1194. IEEE (2018)

Method	**Rick's Cafe**		**Times Square**		**NoLA (Day)**		**NoLA (Night)**	
	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE
Constant velocity	9.6	16.3	15.1	26.9	36.0	70.7	48.4	94.8
Learned Traj (PECNet-Ours)	20.6	29.6	42.7	60.5	35.1	64.1	38.8	67.8
Scene-aware (YNet-Ours)	10.4	16.8	17.3	30.5	30.7	59.2	37.3	71.0
ATP (Finetune)	7.4	10.6	14.7	25.1	30.4	57.1	34.9	62.0
ATP (LC)	10.2	16.5	16.7	29.0	30.3	58.0	36.5	67.8
ATP (LC + Per-Scene FT)	**6.2**	**8.8**	**11.7**	**19.5**	**22.6**	**43.0**	**29.5**	**51.9**

Table 3: Results on four EarthCam scenes. Across all of these challenging in-the-wild scenarios, our ATP method using latent corridors and per-scene finetuning outperforms the baselines and other ATP methods.

Method	ADE	FDE
Scene Aware (YNet-Ours)	32.3	69.1
ATP (Per-Scene Finetune)	22.8	46.6
ATP (LC + Per-Scene Finetune)	**21.8**	**45.5**

Table 4: Results on ETH/UCY. ATP using latent corridors and per-scene finetuning outperforms ATP using fine-tuning alone, and both approaches successfully adapt over the scene-aware baseline.

	SDD		MOTSynth	
Method	ADE	FDE	ADE	FDE
YNet [1]	24.9	49.9	54.4	112.3
YNet-Ours	**17.3**	**33.6**	**47.3**	**96.5**

Table 5: The original YNet model compared to our unimodal-only implementation (YNet-Ours). We see that on both SDD and MOTSynth, YNet-Ours outperforms YNet, and is therefore a stronger baseline.