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

Abstract. Human trajectory prediction is typically posed as a zero-shot generalization problem: a predictor is learnt on a dataset of human motion in training scenes, and then deployed on unseen test scenes. While this paradigm has yielded tremendous progress, it fundamentally assumes that trends in human behavior within the deployment scene are constant over time. As such, current prediction models are unable to adapt to transient human behaviors, such as crowds temporarily gathering to see buskers, pedestrians hurrying through the rain and avoiding puddles, or a protest breaking out. We formalize the problem of context-specific adaptive trajectory prediction and propose a new adaptation approach inspired by prompt tuning called latent corridors. By augmenting the input of a pre-trained human trajectory predictor with learnable image prompts, the predictor improves in the deployment scene by inferring trends from extremely small amounts of new data (e.g., 2 humans observed for 30 seconds). With less than 0.1% additional model parameters, we see up to 23.9% ADE improvement in MOTSynth simulated data and 16.4% ADE in MOT and Wildtrack real pedestrian data. Qualitatively, we observe that latent corridors imbue predictors with an awareness of scene geometry and context-specific human behaviors that non-adaptive predictors struggle to capture.

Keywords: human trajectory prediction, adaptation, test time training, image prompt tuning

1 Introduction

Human motion prediction is a fundamental skill for intelligent systems to effectively navigate the world, assist end-users, and visually survey a scene. To date, learning-based human trajectory prediction has been extensively studied, making huge strides in predicting multimodal future behavior [28], modeling multi-agent social interactions [2, 18, 39], and accounting for scene context [27, 55]. Some approaches have exploited the fact that humans tend to move in similar patterns by storing prior trajectories in a bank and matching a current observed trajectory to the closest stored one [30, 31, 49, 53].

However, a fundamental assumption underlies the current trajectory prediction paradigm: predictors are assumed to be deployed within an *unchanging* context. Nevertheless, in the real world, the context—and thus patterns of human

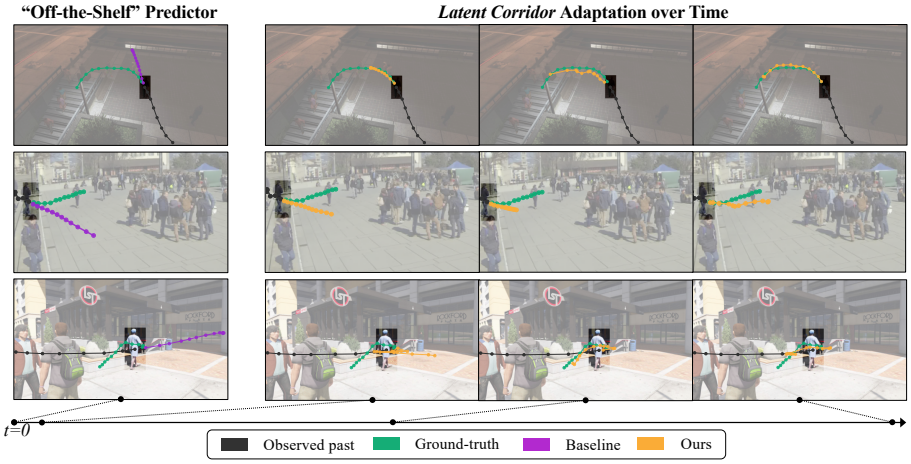

Fig. 1: Adaptive trajectory prediction. (left) Given a history of human behavior (shown in black), the pre-trained predictor $\mathcal{P}$ is unable to understand deployment scene-specific behavior trends, like people entering a subterranean subway entrance (bottom row) or mostly choosing to traverse the staircase as opposed to exploring other parts of the scene at nighttime (top row). (right) When adapting, the number of people and amount of time determine the total number of trajectories observed, and we denote this time-dependent quantity human-seconds. Here, the three columns correspond to our method trained for a very small (left), medium (middle), and large amount of human-seconds (right). Our adaptive *latent corridors* approach enables $\mathcal{P}$ to quickly learn context-specific trends, improving predictions with even small amounts of data, and closing the gap between the ground-truth (green) and predicted behavior (orange). For example, in the middle row, $\mathcal{P}$ predicts the person will move towards the camera, but as our method sees more human-seconds of data, it adapts to the trend that at this point of scene capture in the plaza, people tend to avoid the center of the plaza and instead move diagonally across it.

behavior—will inevitably change over time, as Whyte observed in his famous study of urban public spaces [48]. For example, a new subway entrance being built causes people to descend and exit in new patterns (bottom row, Fig. 1), and icy sidewalks in the winter will be traversed differently than clear ones in the summer. These changes happen on shorter temporal horizons, too: Humans rushing to work during the day will largely ignore each other while nighttime party-goers will walk in cliques. A college campus will be quiet while class is in session and chaotic in the 10 minute period where students rush between classes. Even if data is captured from exactly the same location, human behavior patterns can change in minutes or hours. When faced with such changes, the performance of existing trajectory predictors immediately degrades.

Here, we study how to efficiently *adapt* pre-trained trajectory predictors to observed human behavior within a deployment scene (right, Fig. 1). Our technical approach takes inspiration from the recently popularized paradigm of *prompt-tuning* in large language models [23], but instantiates this idea within the adaptive trajectory prediction problem. Specifically, we augment the input to an

existing human trajectory predictor with a set of latent image prompts, one per each instance of a deployment scene we wish to adapt to. We leave all or most of the predictor frozen, and tune each input prompt with the same trajectory prediction loss that is used to train the base predictor, letting cues in human behaviour be captured while preserving useful structure in the original predictor (e.g., even if a new subway entrance lets people descend into the ground, other people may still move towards the existing above-ground coffee shops). Our adaptation is data-efficient: even extremely small amounts of new human trajectories (e.g., $<$ 1 minute of human seconds, corresponding to 2 humans observed over 30 seconds) can leak sufficient information about how humans interact with the scene (e.g., new subway entrance) or with each other (e.g., nighttime social cliques) to learn this prompt and improve future trajectory prediction. We call our learnable prompt a *latent corridor*: a non-physical corridor that guides human behavior in this scene.

Given the small amounts of data, latent corridors train quickly—on average within minutes on a single 2080 Ti GPU—even without optimizations to improve training speed. With more compute resources and training optimization, latent corridors hold the potential to learn in real time, adapting to transient events such a fire starting or a musician busking. Even with minimal compute and optimization, we also find useful *offline* applications of latent corridors for adaptation to repeated events. For example, imagine analyzing pedestrian flow in a downtown business district — it will differ during the rush hour before work on the weekday and on a weekend morning. During the course of a week, latent corridors could be trained to adapt to the varying scene and context specific behaviours present on different days, times, and surveillance camera viewpoints. Once training is complete, adapted models could be used *online* for months, until an event such as construction starting or a change of season causes a significant change in behaviour. Our method is lightweight, making swapping latent corridors for different times of day or locations inexpensive, even on device.

In our experiments, learning latent corridors for prediction adaptation enables 23.9% prediction accuracy improvement on simulated data from MOT-Synth [12] compared to a non-adaptive predictor that takes as input the scene and has seen all the same training data. On real data from MOT and Wild-Track [7], we see a 16.4% improvement on non-adaptive predictors and an 11.2% improvement with adapting via our method as opposed to just fine-tuning, and on in-the-wild webcam data we see respective improvements of 26.8% and 20.0%. Latent corridor adaptation can improve prediction performance even with a very small amount of data, and continually improves as more data is observed over time.

To summarize, our contributions are as follows:

1. We formalize the adaptive trajectory prediction problem.
2. We propose a novel method for adaptive trajectory prediction by learning lightweight latent corridor prompts in image space, outperforming non-adaptive trajectory predictors with less than a 0.1% parameter increase.

3. We demonstrate qualitative and quantitative improvements of up to 23.9% ADE improvement in MOTSynth simulated data, 16.4% ADE in MOT and Wildtrack real pedestrian data, and 26.8% ADE on in-the-wild webcam data, over a scene-aware baseline.

2 Related Work

Human Trajectory Prediction is a well-studied problem with a long and rich history [36]. Prediction methods started from simple models such as social forces [5, 15], and later added multimodality through Gaussian processes [42]. Recently, modern learning-based predictors have made significant progress in incorporating social interactions between humans modelled via RNNs [17, 39, 44], GANs [13], or conditional VAEs [28], all while generating multimodal future trajectories. Another line of work incorporates the scene context in predictions, getting neural network features from a scene map [20, 22, 29, 30, 37, 43, 51], embedding gridded scene context in an LSTM [29, 52], and more recently using transformers [33, 38]. A few works incorporate scene context by projecting the trajectories into heatmaps so that the network can reason in image space [6, 27], allowing for longer-term predictions. Our work builds upon these advances by adapting an RGB scene-aware pre-trained trajectory predictor [27] over time.

Adapting Human Trajectory Predictors. In the past few years, there has been a growing interest in lightweight ways to modify pre-trained predictors to new data. The key differences between these works lie in 1) *what* they are adapting to, and 2) *how* they adapt. Several works focus on cross-domain transfer, wherein a predictor trained for trajectory prediction in domain A (e.g., New York) is adapted to work in domain B (e.g., London). These works leverage architectures that partition generic trajectory prediction from domain-specific features [46, 50], leverage adaptive meta-learning via Kalman flitering [16], or simply finetune the prediction network on new data [25]. Other works focus on online adaptation over time, for example by adapting to different agent dynamics using recursive least squares [1,10]. A few approaches carry out continual learning by storing embedded past/future trajectory pairs in a memory bank [30, 49, 53] or clustering them [41], predicting by matching an individuals history to the most similar stored past. Going a step further, [31] instead stores representative group trajectories in a scene and during inference, refines the selected trajectory with scene segmentation. In contrast, our novel prompting-based adaptation approach is memory-efficient, using constant extra parameter space regardless of trajectory dataset size, and inherently utilizes the scene segmentation, not requiring extra refinement steps. It also directly augments an existing model as opposed to requiring a new, specialized architecture.

Prompt Tuning in Language & Vision. With the rise of large pre-trained models, efficient adaptation for downstream tasks has gained increasing interest. In the large language model domain, prompt tuning—wherein a few tunable tokens (i.e., prompt) are prepended to input text and tuned per downstream task—has been shown to be remarkably effective [23, 24, 26]. In vision models,

image prompts [45] have been introduced to adapt a vision model to new tasks [3] or instruct an inpainting model to carry out various computer vision tasks [4]. Recently, there have also been attempts at visual prompt tuning, showing that large vision transformers can benefit from utilizing tuned prompts for continual learning or transfer learning [19, 34, 40, 47]. While our task is different, we were inspired by the prompting scheme of [19] that learned classification on a new dataset by introducing a trainable input space prompt and tuning the prompt along with the transformer predictor head. We use visual prompts as a lightweight way of adapting a trajectory predictor. To our knowledge, we are the first work to apply prompt tuning to human trajectory prediction.

3 Problem Formulation

Here, we present a formalization of our proposed adaptive trajectory prediction problem. We seek to generate accurate future trajectories of N agents in a scene. At any time τ, let $\mathbf{o}_\tau := o_{\tau-H:\tau} \in \mathcal{O}$ be the H-step history of observations input into the predictor, $\mathbf{y}_\tau := y_{\tau+1:\tau+T} \in \mathcal{Y}$ be the ground-truth T-step future trajectory, and $\hat{\mathbf{y}}_\tau := \hat{y}_{\tau+1:\tau+T} \in \mathcal{Y}$ be the prediction outputs. We assume access to a base predictor $\mathcal{P}$ pre-trained on a human motion dataset consisting of past and ground-truth future trajectories, $\mathcal{D} = \{\mathbf{o}_i, \mathbf{y}_i\}_{i=1}^K$ to minimize a loss function on trajectories, ℓ (e.g., mean squared error).

Adaptive Trajectory Prediction (ATP). We aim to adapt $\mathcal{P}$ to a deployment scene by observing new human trajectory data over a time interval. Let the dataset of *new* human trajectory data be $\mathcal{D}'_{0:T}$ collected over the T step time interval. Here, $\mathcal{D}'$ can contain data from real or synthetic deployment scenes, a new physical environment or one from the pre-training dataset, a new or previously seen time period (e.g., nighttime vs. daytime), and can be obtained by a new or previously seen camera pose. However, we assume that human pedestrians are present in the scene and they are navigating an outdoor environment. We adapt to scenes captured for a similar amount of time, between $T = 45$ seconds and 5 minutes. We aim to adapt to scene-specific characteristics and fluctuating events beyond those that can be exploited by conditioning only on a segmentation map.

In the adaptive trajectory prediction problem, a subset of the deployment dataset $\mathcal{D}'_{0:t}$, $t < T$ is observed. The goal is create an adapted model $\mathcal{A}[\mathcal{P}]$ such that the prediction error decreases over the future observed deployment data $\mathcal{D}'_{t:T}$:

$$\ell\big(\mathcal{A}[\mathcal{P}(\mathbf{o})], \mathbf{y}\big) < \ell\big(\mathcal{P}(\mathbf{o}), \mathbf{y}\big), \quad (\mathbf{o}, \mathbf{y}) \in \mathcal{D}'_{t:T}. \tag{1}$$

Intuitively, as the value of t increases and the adapted predictor sees more data, as long as the context within the window $[0 : T]$ is consistent, the performance of $\mathcal{A}[\mathcal{P}]$ should improve over the pre-trained predictor $\mathcal{P}$. At deployment, $\mathcal{A}[\mathcal{P}]$ should have learned scene context-specific characteristics of human behavior.

4 Adaptation via Learned Latent Corridors

To adapt our trajectory predictor efficiently, we take inspiration from language-based prompt-tuning [23] and instantiate it within our adaptive trajectory prediction problem. Specifically, we augment a frozen base predictor with a learnable

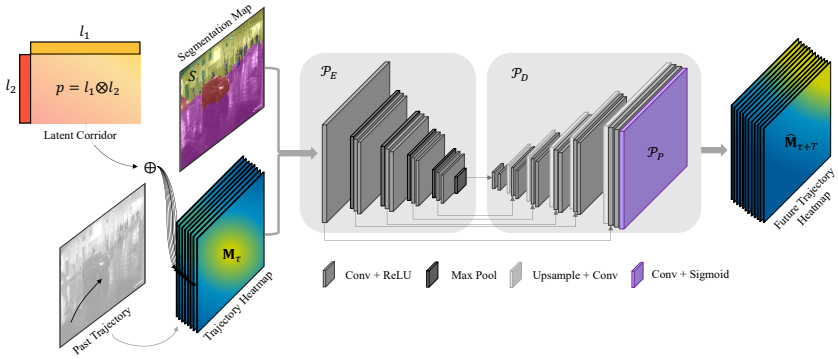

Fig. 2: Adapting a predictor $\mathcal{P}$ with latent corridors. $\mathcal{P}_E$, $\mathcal{P}_D$ and $\mathcal{P}_P$ are pre-trained on the task of human trajectory prediction, taking as input trajectory heatmaps $\mathbf{M}_{\tau-H:\tau}$ and segmentation S, and outputting predicted trajectory heatmaps $\mathbf{M}_{\tau+1:\tau+T}$. We augment $\mathcal{P}$ with a per-scene *latent corridor* p which is summed element-wise to the input trajectory heatmaps. The latent corridors are trained with $\mathcal{P}_E$ and $\mathcal{P}_D$ frozen. The predictor head $\mathcal{P}_P$ can be frozen, tuned on a single deployment scene, or tuned jointly across multiple scenes.

latent prompt we call a *latent corridor*, a non-physical corridor that informs the predictor of human behavior patterns in a specific deployment scene. In this section, we detail our predictor architecture, prompt representation and training, and investigate a suite of latent corridor prompting configurations.

4.1 Base Trajectory Predictor Architecture

Our base predictor model, $\mathcal{P}$, consists of three modules: an encoder $\mathcal{P}_E$, decoder $\mathcal{P}_D$, and predictor head $\mathcal{P}_P$ (see Figure 2). We use the YNet encoder for $\mathcal{P}_E$, and trajectory decoder architecture for $\mathcal{P}_D$ and $\mathcal{P}_P$ [27].

Scene and trajectory representation. To capture the scene, the RGB image I of the first video frame is processed with the Mask2Former semantic segmentation model [8, 9]. The final semantic segmentation map, S, is obtained by downsampling the segmentation classes to $C = 12$ meaningful classes for pedestrians in outdoor environments. We follow [27], and convert the history of observed agent positions $x_{\tau-H:\tau} \in \mathbb{R}^{H\times 2}$ into a trajectory of heatmaps, $\mathbf{M}_{\tau-H:\tau}$, each of the same spatial size as I, for a total size of $H \times h \times w$. Heatmaps are concatenated with the semantic segmentation map S; the final input into $\mathcal{P}$ is $\mathbf{o}_\tau := ([\mathbf{M}_{\tau-H:\tau}, S])$ of size $(C + H) \times h \times w$. The predictor also outputs a trajectory of heatmaps, $\hat{\mathbf{M}}_{\tau+1:\tau+T}$. The corresponding ground truth T-step future agent trajectory, $\mathbf{y}_\tau \in \mathbb{R}^{T\times 2}$, is also converted into a trajectory of heatmaps, $\mathbf{M}_{\tau+1:\tau+T}$, for loss computation.

Loss function. We train the predictor with a binary cross entropy loss on the trajectory heatmaps,

$$\ell := \sum_\tau \mathrm{BCE}(\mathcal{P}([\mathbf{M}_{\tau-H:\tau}, S]), \mathbf{M}_{\tau+1:\tau+T}). \tag{2}$$

For evaluation, a `softargmax` operation is used to sample 2D points from the predicted heatmap $\hat{\mathbf{M}}_{\tau+1:\tau+T}$ as in [27]. This yields predictions in x-y pixel space, $\hat{\mathbf{y}}_\tau = \hat{y}_{\tau+1:\tau+T}$, for computing displacement error metrics in Section 6 with respect to the ground-truth future positions, $\mathbf{y}_\tau$.

4.2 Representing & Learning Latent Corridors

Our key idea for adapting trajectory predictors is to augment the frozen base model $\mathcal{P}$ with a set of trainable prompts, called *latent corridors*, which learn new trends in how humans interact with the scene or with each other. For effective adaptation, we seek two key properties for our latent corridors: parameter-efficient (i.e., we want to minimize the number of parameters that are tuned from new deployment data) and spatially scene-grounded (i.e., the latent should have pixel-wise alignment with the scene image). We first outline our prompting approach, and then discuss a compact but spatially-grounded representation of the prompt that is amenable to parameter-efficient learning.

Latent corridor prompt. For each of the K deployment scenes, we introduce a unique trainable prompt, $p_k \in \mathbb{R}^{h \times w}$, of the same spatial size as the image I that is input into the predictor (left, Fig. 2). For a network that has adapted to K scenes, we have a set K of prompts $p := \{p_0, p_1, \ldots, p_K\}$. Thus, our adaptation rule to scene k is

$$\mathcal{A} := \mathbf{M}_t \oplus p_k \quad \forall t \in \{\tau - H, \ldots, \tau\}. \tag{3}$$

The prompt is summed element-wise to each of the input heatmaps corresponding to the observed trajectories; let $\widetilde{\mathbf{M}} = \mathbf{M} \oplus p$ denote this for any original $\mathbf{M}$. See Supplement Sec.1 for an ablation on prompt location. The adapted predictor takes as input $\mathcal{P}([\widetilde{\mathbf{M}}_{\tau-H:\tau}, S])$.

A compact but spatially-grounded representation. In Equation (3), the prompt is assumed to be the same size as the image, $h \times w$, which is nice for spatial alignment between the prompt and scene. However, in our experiments, where all images are of size $h = 288$ and $w = 480$, this naive image-based representation requires learning an additional 138K parameters. Considering that our base predictor $\mathcal{P}$ has $\sim$ 900K trainable parameters, this prompt increases the model parameter size by over 15%. Instead, we propose a *low-rank representation* of the prompt with rank 1. We initialize our latent corridor as a vector of dimension $h + w$ using Kaiming initialization [14], and parameterize the full prompt as the outer product of the h and w dimensional vectors. This lightweight representation preserves the spatial relationship between the prompt, scene, and trajectory heatmaps and is significantly more compact: it increases the model parameter size by less than 0.1%. We find empirically that this rank 1 matrix performs similarly to a full rank matrix.

Training. We learn the prompt p while the predictor encoder $\mathcal{P}_E$ and decoder $\mathcal{P}_D$ are frozen. The predictor head $\mathcal{P}_P$ is optionally tuned and the latent corridors can be trained individually on one scene at a time, or simultaneously amongst many scenes (see Section 4.3). We use the same trajectory loss ℓ from Equation (2) that the base predictor was pre-trained with.

4.3 Prompting configurations

One of the strengths of our latent corridors approach is that it is compatible with a suite of deployment desiderata that can inform how the prompts are incorporated into the predictor. We identify and study three settings. In each setting, K unique prompts are trained on K deployment scenes individually, but the treatment of the predictor head differs.

1. **Latent corridor adaptation (LC):** The simplest use of latent corridors is to keep the entire base predictor frozen, including the original predictor head $\mathcal{P}_P$. This design is the fastest to train since it has the smallest number of trainable parameters, so it is desirable when rapid adaptation to short-term transient events occurs. It also enables easy recovery of the base predictor model and its original performance by simply not inputting a prompt.
2. **Multi-scene finetuning (LC + Joint FT):** In this setting, each of the K deployment scenes has a unique latent, but one single predictor head $\mathcal{P}_P$ is jointly tuned across the K scenes. The latents retain unique scene information, but $\mathcal{P}$ is better adapted to in conjunction with the per-scene latents. Similarly to the LC approach above, this is a more compact configuration since one prediction model is used for all scenes, so it is desirable if deployment hardware has limited space. It is also faster to train than per-scene finetuning especially as K grows, since multiple predictor heads do not have to be tuned.
3. **Per-scene finetuning (LC + Per-Scene FT):** If one seeks to maximize performance within a *specific* deployment scene, one can *jointly* finetune K predictor heads $\mathcal{P}_{P_k}$ with K latent corridors (one per scene). While this results in the need for a unique predictor for each deployment scene, we find empirically that this method achieves best in-scene adaptation performance.

5 Experimental Setup

We study our approach on synthetic and real datasets. Here, we detail these datasets, describe how we evaluate adaptation quality over time, and outline our trajectory predictor baselines. Together, our experiments on MOT, WildTrack and EarthCam scenes cover a diverse range of key real-world properties including different lighting conditions, flat vs varied environment topologies, different crowd densities, and a variety of types of scenes.

5.1 Datasets

MOTSynth. We start in simulation with MOTSynth [12], a synthetic pedestrian detection and tracking dataset of over 700 90-second videos with varying camera viewpoints and outdoor environments. Pedestrians carry out simple actions such as walking, standing, or running, and follow manually pre-planned flows as well as a collision avoidance algorithm. MOTSynth has over 17 hours of video; we select a subset of approximately 11 hours of video corresponding

to the 437 scenes with a static camera. Due to the large size of the dataset and perfect ground truth detections, MOTSynth was a good starting point.

MOT & WildTrack. We also evaluated our approach on the real-world pedestrian datasets MOT and WildTrack. WildTrack [7] consists of 7 static camera viewpoint videos of pedestrians walking through a plaza. We randomly selected 3 videos and took the first 5 minutes of each video. MOT, the multiple object tracking benchmark, consists of several video datasets of pedestrians navigating outdoor environments. We combined the datasets MOT15 [21], MOT16 [32], and MOT20 [11], and removed videos with dynamic cameras, pedestrian density of less than 10, and length less than 45s, leaving 4 scenes: MOT16-03, MOT20-02, AVG-TownCentre, and PETS09-S2L2.

EarthCam. We evaluate our approach in-the-wild on data from `https://www.earthcam.com`, which has livestream webcam data from around the world available, and from which we scrape four 5 minute segments of data.

5.2 Train-Test Split Over Time in Human Seconds

Given a video of pedestrians in motion, we prepare our dataset to adapt to the scene as follows. We use provided annotations or run ByteTrack [54] to get tracklets of all N detectable identities in the video. Tracklets are downsampled to 2 timesteps per second, then windowed by 20 timesteps with an observed length $H = 8$ and future length $T = 12$. Then, we sort the N identities chronologically by their first appearance in the scene. We select the first 80% of identities that appear in the scene for our training dataset, and hold out the last 20% for the testing set. When we conduct experiments over time, the testing set is always the same 20% of identities, whereas the training set consists of the first m% of identities, $m \leq 80\%$. We report adaptation time in human-seconds of observation, where 1 person for 1 second is 1 human-second, so for instance, 30 people observed for 10 seconds each would be 300 human-seconds.

5.3 Base Predictor Pre-Training Dataset

We first pre-train our base predictor $\mathcal{P}$ described in Sec 4.1 on a dataset $\mathcal{D}$ which consists of simulated human pedestrian agents moving around various outdoor environments, captured from a single static RGB camera with any point of view (not necessarily birds eye view). This dataset comes from 437 90-second MOTSynth videos with a static camera. We pre-train our predictor on a train-test split over time of $\mathcal{D}$ as described in the previous section.

5.4 Baselines

We evaluate our method with the following baselines.

1. *Constant velocity*: As in [5].
2. *Learned trajectory*: Simplified PECNet [28], which is a predictor learned from position histories but no scene information.

Method	MOTSynth		MOT		Wildtrack		EarthCam	
	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE
Constant velocity	78.5	160.4	47.7	99.3	44.9	90.1	27.3	52.2
Learned trajectory (PECNet-Ours)	51.2	100.0	49.7	103.4	43.8	83.1	34.3	55.5
Scene-aware (YNet-Ours)	47.3	96.5	44.2	91.0	33.6	67.7	23.9	44.4
ATP (Finetune)	-	-	41.8	86.9	31.7	63.5	21.9	38.7
ATP (LC)	44.6	90.2	43.0	89.2	32.9	65.9	23.4	42.8
ATP (LC + Joint Finetune)	**42.6**	**85.3**	-	-	-	-	-	-
ATP (LC + Per-Scene Finetune)	-	-	**37.4**	**74.3**	**27.4**	**54.7**	**17.5**	**30.8**

Table 1: Our method vs baselines on synthetic and real-world datasets. The average ADE and FDE in pixel space of (left-to-right) 437 MOTSynth scenes, 4 MOT scenes, 3 WildTrack scenes, and 4 EarthCam scenes. Across these synthetic and real deployment scenes, our latent corridor-only adaptation method is comparable to fine-tuning the predictor head and outperforms non-adaptive baselines. Our method with latent corridors and per-scene finetuning consistently outperforms all other approaches.

3. *Scene-aware*: Simplified YNet [27], which is a learned predictor with position and scene input. See Supplement Sec. 3 for a comparison with YNet.
4. *ATP Finetune*: Adaptation baseline which finetunes scene-aware predictor head $\mathcal{P}_P$ without latent corridors.

We implement the learned trajectory baseline by taking the encoder, decoder and trajectory loss from PECNet [28]; we use a successful architecture on the trajectory prediction problem, removing multimodality for simplicity. Similarly, YNet's [27] encoder and trajectory decoder are used for the scene-aware baseline.

5.5 Metrics

Our experiments are evaluated using the Average Displacement Error (ADE) [35] and Final Displacement Error (FDE) [2] metrics. ADE is the average l_2 error between the entire predicted and ground truth trajectory, while FDE is the l_2 error between just the final point of the predicted and ground truth trajectories.

6 Results

Here, we detail quantitative and qualitative results on MOTSynth, MOT, and WildTrack, and EarthCam datasets.

6.1 MOTSynth Results

We first evaluate predictor improvement enabled by latent corridors when the deployment scene is in the pre-training dataset, $\mathcal{D}$. We train the baselines and two variants of our ATP method (LC and LC + Joint Finetune, described in Sec. 4.3) on the MOTSynth pre-training dataset $\mathcal{D}$. All models see the same training data, and both the scene-aware baseline and our adaptive method are conditioned on scene semantic segmentation maps. Results are in the first column of Table 1. The learned trajectory baseline significantly improves over constant velocity, and adding in scene awareness results in a 7.6% performance gain on ADE and 3.5% gain on FDE. Our latent corridor approach results in a 5.7%

Fig. 3: Qualitative results on MotSynth (top; synthetic) and MOT and WildTrack (bottom; real). These examples show scenarios where our LC + per-scene finetune ATP method (orange) outperforms the scene-aware baseline (purple). In several MOTSynth examples, the baseline predicts the pedestrian floats into the air (top row), while our method has gained awareness of where the 3D ground plane lies in the 2D image. We also note that patterns of behaviour such as walking on the sidewalk instead of into the road (second row left) and walking up the traversable portion of stairs (second row right) are captured. On real data, we observe similar awareness of the ground plane and obstacles, as well as a better understanding of nuanced human behavior patterns such as crossing diagonally across a plaza.

and 9.9% improvement over the scene-aware baseline for ADE, without and with joint finetuning of the predictor layer respectively, and a 6.5% and 11.6% improvement on FDE. This indicates that our latent corridor approach more effectively learns scene-conditioned information that is useful for the trajectory prediction task.

Next, for a random subset of 25 of the MOTSynth scenes, we additionally train latent corridors with per-scene finetuning (LC + Per-Scene FT, Sec 4.3). The datasets $\mathcal{D}'_{0:t}$ correspond to increasing human-second lengths. The test set trajectories are the last 20% of agents in the deployment scene, and the training sets consist of the first $2, 4, 8, 16, 32, 48, 64$ and 80% of agents to enter the scene. Results normalized per-scene and averaged are shown in Fig. 5b. Using only latent corridors performs comparably to only finetuning the last layer for each deployment scene, while latent corridors with per-scene finetuning yields significant performance gains. We visualize the ADE results per-scene with our

Fig. 4: Qualitative results over time. Our method's predictions trained on a short number of human seconds (2%) are shown in light orange, to dark orange for a human seconds training time of 80%. With the latent corridor trained on a tiny amount of data, the predictions can significantly improve, but at times are close to the baseline. When more human seconds of data are seen, the adaptation results consistently improve.

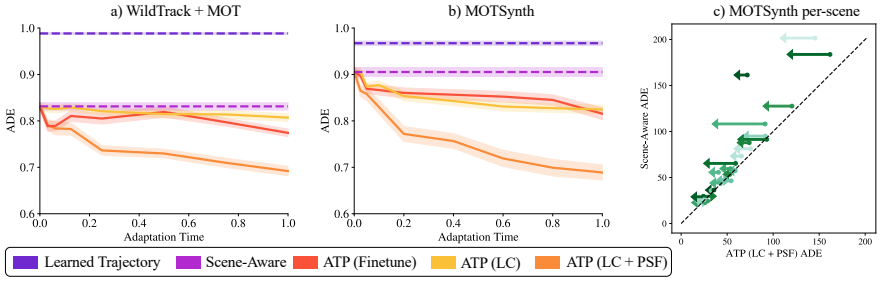

Fig. 5: Learning latent corridors over time. (a and b): The x-axis represents adaptation time in human-seconds, or the amount trajectories used to train $\mathcal{A}(\mathcal{P})$, and the y-axis represents the ADE. Results are normalized per-scene and averaged over models trained on 25 MOTSynth scenes (a) and 7 from MOT and WildTrack (b), with shaded area $\sigma/10$. On all data, even with a short adaptation time, our methods improve on the baselines, and as adaptation time increases, performance improves. Latent corridors + per-scene finetuning has the best performance. c) Comparison to baseline over many MOTSynth scenes for models trained with 8% (point) and 80% (arrowhead) human-second datasets. Each arrow represents one scene, with the ADE using our ATP method plotted against the scene-aware baseline ADE. For some deployment scenes, the scene-aware baseline suffices, whereas other scenes see much more significant benefits from our method.

LC + finetuning method trained with 8% and then trained with 80% of the data in Fig. 5c. We see that while the effectiveness of the adaptation for varying time horizons differs for each deployment scene, there are many scenes where adaptation yields significant gains, and some where the improvement on ADE error is up to 63.4%. We hypothesize that scenes where our method yields smaller gains exhibit behaviors and environment geometries that the prior $\mathcal{P}$ is sufficient for.

Qualitative results. Visualizations from the per-scene models can be seen in Fig. 3 and Fig. 4. We observe many examples where the scene-aware baseline seems to lack awareness of the ground plane. In multiple instances, the scene-aware baseline predicts a pedestrian floats into the air (Fig. 3 top row and Fig. 4), whereas our predictions lie closely in line with the ground truth trajectory in terms of distance from the ground plane. This holds even for non-planar ground planes: When a pedestrian walks up stairs, our model seems to better understand

their 3D structure, for example in the bottom right of Fig. 4. Additionally, our approach captures trends of behaviour in scenes such as walking on the sidewalk instead of into the road, walking around the rotunda, and walking up the section of smaller, easily traversable steps (see Fig. 3 second row, left to right).

6.2 MOT and WildTrack Results

We next investigate if latent corridors help predictor adaptation when the deployment scene is *outside* of the pre-training dataset, and if latent corridors outperform ATP via direct finetuning on deployment data. Specifically, we use the base predictor $\mathcal{P}$ which only saw MOTSynth data $\mathcal{D}$ and then directly do sim-to-real adaptation via latent prompting, finetuning, or a combination thereof using each of seven *real human pedestrian scenes* from MOT and Wildtrack as $\mathcal{D}'$. Plots of the ADE over time averaged over these scenes is shown in Figure 5a. Regardless of the ATP setting, adaptation over time results in a consistent error reduction compared to the baselines. While only finetuning seems to be slightly more effective on real-world scenes than prompting alone, our latent corridors + per-scene finetuning approach is significantly more effective (11.2%) than finetuning alone (see Table 1).

Qualitative results. Visualizations of baselines and our LC + PSF approach on MOT and WildTrack are in Fig. 3. Similar trends from experiments on synthetic data carry over. Our approach enables the predictor to better ground future behavior in the scene geometry: for example, pedestrians are no longer predicted to float upwards (third row, left and middle), and future behavior is guided by a finer-grained awareness of obstacles (third row right, bottom row left and middle). We also observe our adapted predictor learning trends in human behavior: in the WildTrack plaza environment shown in Fig. 1 middle row and Fig. 3 bottom right, pedestrians tend to avoid the middle of the plaza and instead cross it diagonally. Our method learns this subtle, scene-specific behavior pattern and thus predicts more accurately.

6.3 EarthCam Results

Finally, we evaluate the performance of our approach on truly "in-the-wild" data by scraping four 5-minute videos of pedestrian data captured in different locations: Rick's Cafe in Jamaica, Times Square in New York City, and Bourbon Street in New Orleans during daytime and nighttime. We use ATP as in Sec. 6.2. Quantitatively, the results on this in-the-wild data align with our results on real and synthetic data: our ATP model with latent corridors and per-scene finetuning outperforms the non-adaptive baselines and pure finetuning (see Table 1). Qualitatively in Fig. 6, we see a variety of interesting adaptations. At the cafe, there is a complicated path through an overlook that twists down stairs towards the water. The scene-aware baseline often predicts that people will jump over a ledge, whereas our method learns the boundaries of the paths that people follow and is able to predict that people will stay within those boundaries. In Times Square, the scene-aware baseline does not recognize that people on a billboard will stay within the billboard, whereas our method is able to recognize that.

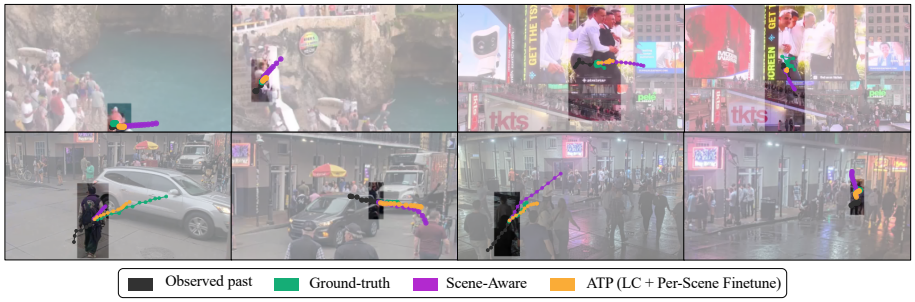

Fig. 6: Results on EarthCam webcam data. At a cafe in Jamaica (top left 2 panels) with an overlook and stairs on the edge of the water, our model (orange) is able to correctly predict that people will stick to moving up and down the path created by the stairs, while the scene-aware baseline (purple) predicts people will walk directly into the water or over the edge of the path. In Times Square, a billboard depicts human actors in motion (top right 2 panels). The scene-aware baseline assumes that these actors will move following their observed history, while our model correctly predicts that anyone in the area of the billboard will stay in the billboard. We train our method on one NoLA intersection during the daytime (bottom left 2 panels), when pedestrians must obey vehicles moving through the street, and at nighttime (bottom right 2 panels), when pedestrians take over. During the day, the model learns to account for humans walking through crosswalks instead of diagonally into the intersection, and to account for transient push-carts in the area.

In the NoLA videos, our method adapts to daytime patterns where pedestrians navigate around carts and don't frequently walk diagonally through streets because of through-traffic, but when the adaptive predictor observes nighttime behavior, it no longer has to respect these daytime patterns and learns to predict pedestrians as crossing diagonally. Similarly to the prior datasets, our ATP model again learns to ground pedestrian future behavior in the 3D ground plane.

7 Conclusion

In this work, we formalize and study the problem of adaptive trajectory prediction: the ability of human predictors to adapt to changing deployment conditions and environments. To this end, we proposed a lightweight adaptation approach grounded in image-based prompt tuning called latent corridors. Through extensive experiments on both simulated and real-world pedestrian datasets, we observed that latent corridors enable a data-efficient way to adapt pre-trained predictors to new deployment-scene-specific human behavior.

8 Acknowledgements

We thank Anastasios Angelopoulos, Antonio Loquercio, and Jathushan Rajasegaran for useful discussions and feedback. This work was supported by ONR MURI N00014-21-1-2801 and a NSF Graduate Fellowship.

References

1. Abuduweili, A., Li, S., Liu, C.: Adaptable human intention and trajectory prediction for human-robot collaboration. arXiv preprint arXiv:1909.05089 (2019)
2. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: Human trajectory prediction in crowded spaces. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 961–971 (2016)
3. Bahng, H., Jahanian, A., Sankaranarayanan, S., Isola, P.: Exploring visual prompts for adapting large-scale models. arXiv preprint arXiv:2203.17274 (2022)
4. Bar, A., Gandelsman, Y., Darrell, T., Globerson, A., Efros, A.A.: Visual prompting via image inpainting. arXiv preprint arXiv:2209.00647 (2022)
5. Best, R., Norton, J.: A new model and efficient tracker for a target with curvilinear motion. IEEE Transactions on Aerospace and Electronic Systems **33**(3), 1030–1037 (1997). https://doi.org/10.1109/7.599328
6. Cao, Z., Gao, H., Mangalam, K., Cai, Q.Z., Vo, M., Malik, J.: Long-term human motion prediction with scene context. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16. pp. 387–404. Springer (2020)
7. Chavdarova, T., Baqué, P., Bouquet, S., Maksai, A., Jose, C., Bagautdinov, T., Lettry, L., Fua, P., Van Gool, L., Fleuret, F.: Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5030–5039 (2018)
8. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation (2022)
9. Cheng, B., Schwing, A.G., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation (2021)
10. Cheng, Y., Zhao, W., Liu, C., Tomizuka, M.: Human motion prediction using semi-adaptable neural networks. In: 2019 American Control Conference (ACC). pp. 4884–4890. IEEE (2019)
11. Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., Leal-Taixé, L.: Mot20: A benchmark for multi object tracking in crowded scenes. arXiv preprint arXiv:2003.09003 (2020)
12. Fabbri, M., Brasó, G., Maugeri, G., Cetintas, O., Gasparini, R., Ošep, A., Calderara, S., Leal-Taixé, L., Cucchiara, R.: Motsynth: How can synthetic data help pedestrian detection and tracking? In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10849–10859 (2021)
13. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social gan: Socially acceptable trajectories with generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2255–2264 (2018)
14. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034 (2015)
15. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. Physical review E **51**(5), 4282 (1995)
16. Ivanovic, B., Harrison, J., Pavone, M.: Expanding the deployment envelope of behavior prediction via adaptive meta-learning. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 7786–7793. IEEE (2023)
17. Ivanovic, B., Pavone, M.: The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2375–2384 (2019)

18. Ivanovic, B., Schmerling, E., Leung, K., Pavone, M.: Generative modeling of multimodal multi-human behavior. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 3088–3095. IEEE (2018)
19. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. In: European Conference on Computer Vision. pp. 709–727. Springer (2022)
20. Kitani, K.M., Ziebart, B.D., Bagnell, J.A., Hebert, M.: Activity forecasting. In: Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV 12. pp. 201–214. Springer (2012)
21. Leal-Taixé, L., Milan, A., Reid, I., Roth, S., Schindler, K.: Motchallenge 2015: Towards a benchmark for multi-target tracking. arXiv preprint arXiv:1504.01942 (2015)
22. Lee, N., Choi, W., Vernaza, P., Choy, C.B., Torr, P.H., Chandraker, M.: Desire: Distant future prediction in dynamic scenes with interacting agents. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 336–345 (2017)
23. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691 (2021)
24. Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190 (2021)
25. Li, Y., Zhao, S.Z., Xu, C., Tang, C., Li, C., Ding, M., Tomizuka, M., Zhan, W.: Pre-training on synthetic driving data for trajectory prediction. arXiv preprint arXiv:2309.10121 (2023)
26. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM Computing Surveys **55**(9), 1–35 (2023)
27. 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)
28. Mangalam, K., Girase, H., Agarwal, S., Lee, K.H., Adeli, E., Malik, J., Gaidon, A.: It is not the journey but the destination: Endpoint conditioned trajectory prediction. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. pp. 759–776. Springer (2020)
29. Manh, H., Alaghband, G.: Scene-lstm: A model for human trajectory prediction. arXiv preprint arXiv:1808.04018 (2018)
30. Marchetti, F., Becattini, F., Seidenari, L., Bimbo, A.D.: Mantra: Memory augmented networks for multiple trajectory prediction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7143–7152 (2020)
31. 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)
32. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: Mot16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831 (2016)
33. Ngiam, J., Caine, B., Vasudevan, V., Zhang, Z., Chiang, H.T.L., Ling, J., Roelofs, R., Bewley, A., Liu, C., Venugopal, A., et al.: Scene transformer: A unified architecture for predicting multiple agent trajectories. arXiv preprint arXiv:2106.08417 (2021)

34. Nie, X., Ni, B., Chang, J., Meng, G., Huo, C., Xiang, S., Tian, Q.: Pro-tuning: Unified prompt tuning for vision tasks. IEEE Transactions on Circuits and Systems for Video Technology (2023)
35. Pellegrini, S., Ess, A., Schindler, K., Van Gool, L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In: 2009 IEEE 12th international conference on computer vision. pp. 261–268. IEEE (2009)
36. Rudenko, A., Palmieri, L., Herman, M., Kitani, K.M., Gavrila, D.M., Arras, K.O.: Human motion trajectory prediction: A survey. The International Journal of Robotics Research **39**(8), 895–935 (2020)
37. Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofighi, H., Savarese, S.: Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1349–1358 (2019)
38. Salzmann, T., Chiang, L., Ryll, M., Sadigh, D., Parada, C., Bewley, A.: Robots that can see: Leveraging human pose for trajectory prediction. IEEE Robotics and Automation Letters **8**(11), 7090–7097 (2023). https://doi.org/10.1109/LRA.2023.3312035
39. Salzmann, T., Ivanovic, B., Chakravarty, P., Pavone, M.: Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16. pp. 683–700. Springer (2020)
40. Sandler, M., Zhmoginov, A., Vladymyrov, M., Jackson, A.: Fine-tuning image transformers using learnable memory. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12155–12164 (2022)
41. Sun, J., Li, Y., Fang, H.S., Lu, C.: Three steps to multimodal trajectory prediction: Modality clustering, classification and synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13250–13259 (2021)
42. Tay, M.K.C., Laugier, C.: Modelling smooth paths using gaussian processes. In: Field and Service Robotics: Results of the 6th International Conference. pp. 381–390. Springer (2008)
43. Varshneya, D., Srinivasaraghavan, G.: Human trajectory prediction using spatially aware deep attention models. arXiv preprint arXiv:1705.09436 (2017)
44. Vemula, A., Muelling, K., Oh, J.: Social attention: Modeling attention in human crowds. In: 2018 IEEE international Conference on Robotics and Automation (ICRA). pp. 4601–4607. IEEE (2018)
45. Wang, J., Liu, Z., Zhao, L., Wu, Z., Ma, C., Yu, S., Dai, H., Yang, Q., Liu, Y., Zhang, S., et al.: Review of large vision models and visual prompt engineering. arXiv preprint arXiv:2307.00855 (2023)
46. Wang, L., Hu, Y., Sun, L., Zhan, W., Tomizuka, M., Liu, C.: Transferable and adaptable driving behavior prediction. arXiv preprint arXiv:2202.05140 (2022)
47. Wang, Z., Zhang, Z., Lee, C.Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., Pfister, T.: Learning to prompt for continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 139–149 (2022)
48. Whyte, W.H.: The social life of small urban spaces. Washington, D.C.:Conservation Foundation (1980)
49. Xu, C., Mao, W., Zhang, W., Chen, S.: Remember intentions: Retrospective-memory-based trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6488–6497 (2022)

50. Xu, Y., Wang, L., Wang, Y., Fu, Y.: Adaptive trajectory prediction via transferable gnn. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6520–6531 (2022)
51. 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 (2018). https://doi.org/10.1109/WACV.2018.00135
52. 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)
53. Yang, B., Fan, F., Ni, R., Li, J., Kiong, L., Liu, X.: Continual learning-based trajectory prediction with memory augmented networks. Knowledge-Based Systems **258**, 110022 (2022)
54. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., Wang, X.: Bytetrack: Multi-object tracking by associating every detection box (2022)
55. Zhao, T., Xu, Y., Monfort, M., Choi, W., Baker, C., Zhao, Y., Wang, Y., Wu, Y.N.: Multi-agent tensor fusion for contextual trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12126–12134 (2019)

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,

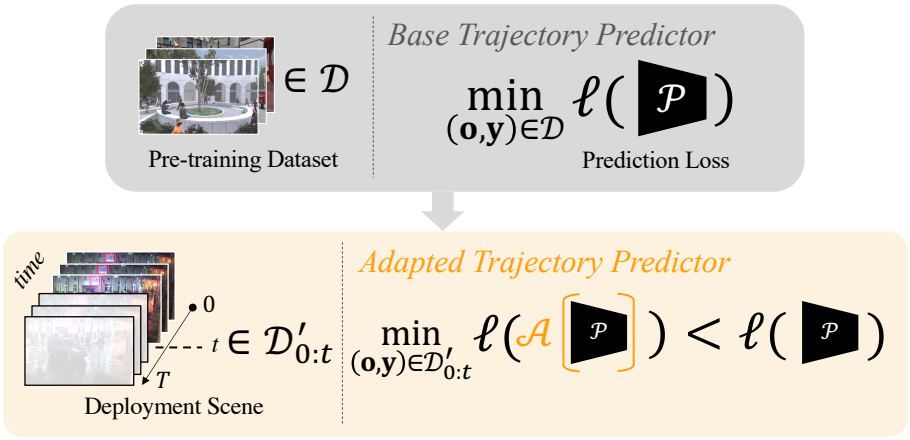

Fig. 1: Adaptive trajectory prediction. ATP, formulated in Sec. 3, allows a pre-trained predictor $\mathcal{P}$ to adapt to a new deployment scene by learning over time on the deployment scene. Once adaptation has occurred, the adapted predictor $\mathcal{A}[\mathcal{P}]$ should perform better on the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

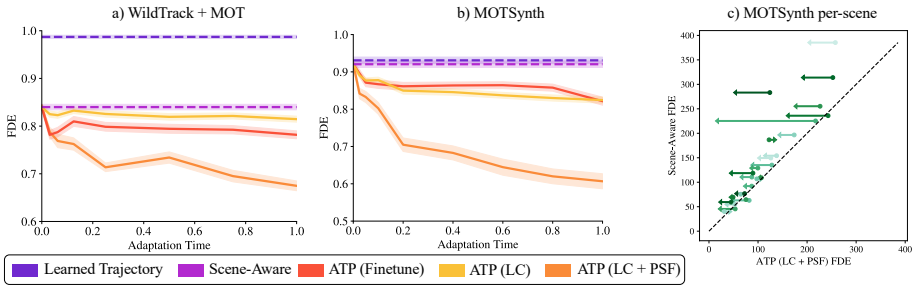

Fig. 2: Adaptation over time FDE. As in main text Fig. 5, the x-axis represents normalized adaptation time in human-seconds. The y-axis represents the FDE (lower is better). Results are normalized per-scene and averaged over models trained on 25 MOTSynth scenes (a) and 7 from MOT and WildTrack (b), with 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

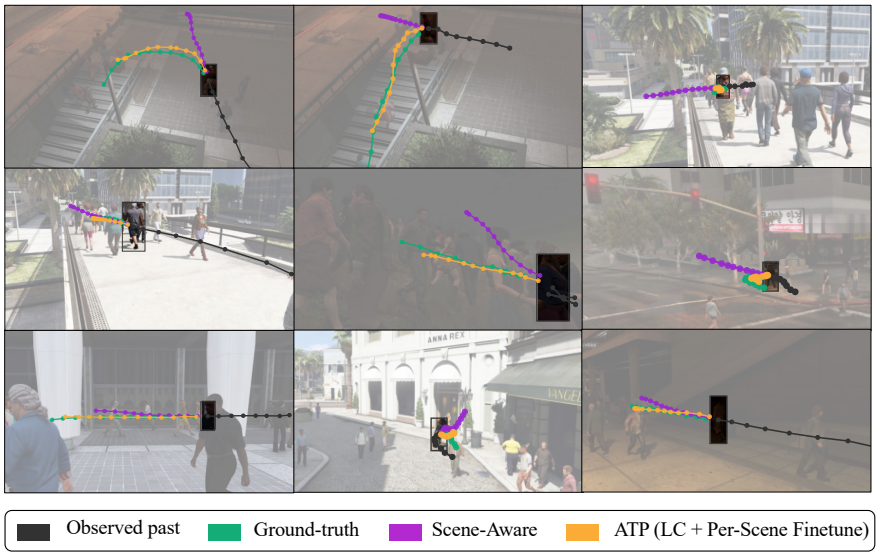

Fig. 3: Additional MOTSynth qualitative results. (top left and middle) From several examples of pedestrians in motion, our approach (orange) is able to learn that in this scene, most pedestrians will turn to go down the stairs, while the scene-aware baseline (purple) struggles to understand this scene-specific 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.