Hexa-P: Physical Plausibility-aware Pre-training for Pedestrian Path Prediction

Hiromu Taketsugu[1*], Shohei Nobuhara[2] and Norimichi Ukita[1]

[1*]Toyota Technological Institute, 2-12-1, Hisakata, Tenpaku-ku, Nagoya, 468-8511, Japan.
[2]Kyoto Institute of Technology, 1, Hashigami-cho, Matsugasaki, Sakyo-ku, Kyoto, 606-8585, Japan.

*Corresponding author(s). E-mail(s): sd25502@toyota-ti.ac.jp;
Contributing authors: nob@kit.ac.jp; ukita@toyota-ti.ac.jp;

Abstract

Pedestrian path prediction is a safety-critical component of autonomous vehicles and social robots that must navigate alongside humans in open urban environments. In such deployments, predictions that imply unnatural accelerations or impossible turns can directly trigger unsafe planning decisions, especially when only a few past frames of the pedestrian are observed. Because pedestrians are full-body humans whose paths are governed by articulated locomotion, we exploit two resources: 3D human body pose, which encodes locomotion-level constraints, and a humanoid physics simulator, which synthesizes virtually unlimited, physically executable pose-and-path data. We propose "Physical Plausibility-aware Pre-training for Pedestrian Path Prediction" (Hexa-P), which uses this synthesized data to jointly pre-train a path predictor and a plausibility estimator. Unlike prior work that uses a separately trained plausibility estimator, this joint pre-training improves both prediction accuracy and physical plausibility. While this simulation involves only a single agent, multiple agents mutually influence one another's future paths in real-world scenes. We therefore introduce an adapter-based fine-tuning that preserves physical knowledge while adapting to scene-specific social path interactions. We also propose Target-relative Positional Encoding (TaPE) to bridge the gap between single-agent simulation and multi-agent path prediction. Experiments on JRDB, JTA, and the Urban dataset demonstrate accurate and plausible predictions of Hexa-P. Particularly in safety-critical short-observation scenarios, Hexa-P reduces prediction errors by up to 50% on the JRDB dataset compared with a strong pose-aware baseline.

Keywords: Path Prediction, Embodied AI, Physics Simulation, Pre-training, Sim-to-Real Transfer

1 Introduction

Pedestrian path prediction is a safety-critical component of autonomous-driving and social-robot-navigation systems in open urban environments [2–4]. Since these predictions are used as input to motion planning, physically implausible predictions, such as those with impossible turns or unnatural accelerations, can lead to unsafe steering or braking decisions.

Deep learning methods [5–8] achieve substantial accuracy when sufficient past observations are available, but produce physically implausible predictions (Fig. 1 (a)) when the observation history is limited, *e.g.*, when a pedestrian suddenly appears from behind an obstacle.

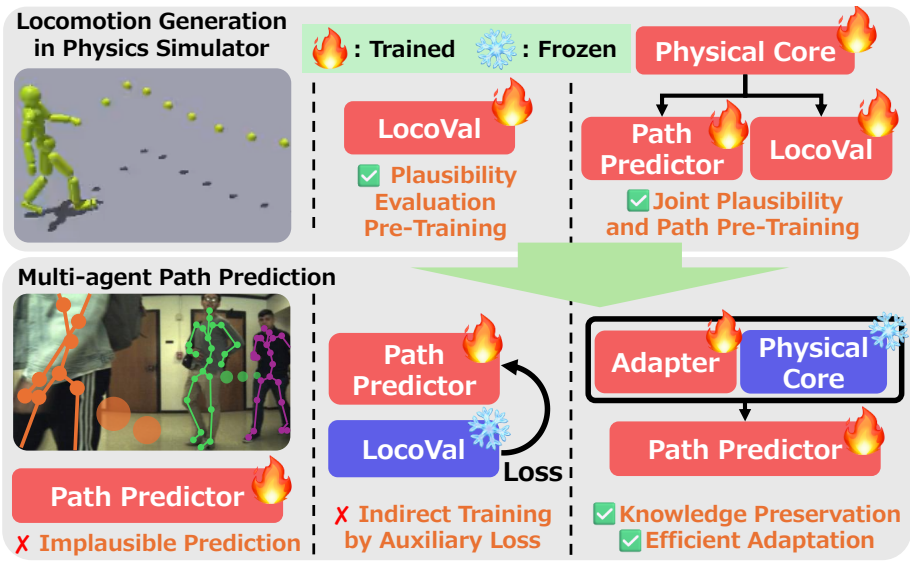


Fig. 1: Comparison of path prediction approaches. (a) Prior work lacks physical plausibility. (b) EmLoco [1] employs physics simulation only for auxiliary loss-based training. (c) Hexa-P jointly pre-trains path prediction and plausibility estimation, then fine-tunes with adapters to preserve physical knowledge while adapting to multi-agent scenarios

Why do these methods fail? Classical models such as the Social Force Model [9, 10] and kinematic models [11, 12] use velocity and acceleration cues, but represent each pedestrian as a 2D point and ignore the physics of full-body locomotion. A 2D point sequence is insufficient because two pedestrians with identical past paths can take different future paths, and short observations remain ambiguous despite recent methods designed to handle them [13, 14]. The 3D body pose, however, can resolve this ambiguity; for example, a body oriented to the right is likely to turn right.

While several methods [4, 7, 15, 16] use 3D human poses to model the pose-path relationship, their applicability is limited by the scarcity of large-scale datasets with accurate 3D pose annotations. Although motion capture systems can collect accurate 3D poses, most motion capture datasets contain only a single agent per recording, which is insufficient because social interactions are essential for path prediction in real-world scenes.

To address this data-collection problem, we decompose it into two components: *physical constraints* governed by individual human locomotion and *social interactions* among multiple agents. This decomposition is possible because physical plausibility requires accurate 3D poses, whereas social interactions can be assessed without them.

Based on this decomposition, we propose **P**hysical **P**lausibility-aware **P**re-training for **P**edestrian **P**ath **P**rediction (**Hexa-P**), illustrated in Fig. 1 (c). In the pre-training stage, a humanoid physics simulator generates a large number of 3D pose-path pairs that satisfy physical constraints. We use these pairs to jointly pre-train a path predictor and a plausibility estimator. Unlike prior work (Fig. 1 (b)) that trains the plausibility estimator separately and uses it as an auxiliary loss [1], our joint pre-training improves both prediction accuracy and physical plausibility. The pre-trained models are then fine-tuned on real-world data to incorporate social interactions among multiple agents.

Our contributions are summarized as follows:

1. **Physical plausibility-aware pre-training.** We jointly pre-train a path predictor and a plausibility estimator inside a humanoid physics simulator, which provides virtually unlimited 3D pose-path data that satisfies physical constraints.

2. **Adapter-based fine-tuning.** We freeze the pre-trained Physical Core and train only lightweight adapters on real-world data, preserving the simulator-learned physical knowledge while adapting to multi-agent interactions.
3. **Target-relative Positional Encoding (TaPE).** We encode each agent's past trajectory in its own coordinate frame and the inter-agent relative positions separately, which bridges single-agent simulation and multi-agent prediction.

With these contributions, Hexa-P achieves state-of-the-art accuracy, reducing errors by up to 50% in short-observation scenarios where pose-path priors are most valuable.

2 Related Work

2.1 Deep Learning-based Pedestrian Path Prediction

Early RNN-based methods [5, 6, 17] model only temporal dependencies between past and future paths, while several methods [8, 11, 18, 19] introduce graph-based spatio-temporal interaction modeling. Recent attention-based approaches [20–24] improve the spatio-temporal modeling. However, these methods treat pedestrians simply as 2D point sequences, leading to physically implausible path predictions.

2.2 Pose-Based Path Prediction

Several works incorporate human 3D poses to model physical constraints [2, 15, 16, 25]. While these methods take only fixed-length observations, mask-based prompting in Social-Transmotion (Social-Trans) [7] enables variable-length observation inputs and can handle diverse visual cues, such as 2D paths and 3D poses. Recent advances include pose-knowledge distillation [26], joint path and pose prediction [27, 28], and generic pose encoders optimized for path prediction [4].

However, these pose-based methods face data scarcity: learning pose-path relationships requires large-scale diverse data with accurate 3D pose annotations, which is expensive to collect. This data scarcity problem motivates our physical plausibility-aware pre-training approach on physics simulators.

2.3 Physics-Based Locomotion Generation

Physically plausible motions can be generated through physics simulation of humanoid locomotion. DeepMimic [29] introduces motion imitation for physics-based humanoid control. AMP [30] uses adversarial training to generate natural, diverse humanoid motions. PACER [31] generates natural humanoid locomotion given conditioned future paths.

While these methods use physics simulation only for animation or robotics, Hexa-P leverages physics-based locomotion generation for pedestrian path prediction. Simulators can generate unlimited training data with ground-truth synchronized 2D paths and 3D poses, enabling sufficient learning of pose-path relationships in terms of their physical plausibility.

2.4 Physics Simulation-based Path Prediction

EmLoco [1] makes the first attempt to bridge physics simulation and path prediction. However, EmLoco relies on a two-step learning strategy: it first learns a surrogate plausibility function (LocoVal) in a physics simulator, then uses the pre-trained LocoVal as an auxiliary loss (EmLoco loss) to train the path predictor on real-world data. This indirect plausibility evaluation via LocoVal yields only an approximation.

In Hexa-P, by contrast, both pose-path plausibility and path prediction are initially learned jointly in a physics simulator. This joint learning without approximation allows for better physics-aware pose-path learning.

2.5 Future Prediction for Autonomous Navigation

Safe navigation among humans requires predicting how the surrounding scene will change, and prior work addresses this at different levels. Crossing-intention prediction estimates whether a pedestrian will cross the road [32–34]. Occupancy-based methods predict a grid of free and occupied space over time [35, 36]. Pedestrian path prediction, which is the task we address, instead predicts agents' future paths.

Path prediction is also used in generative driving simulation. Trajectory-conditioned world

models [37, 38] take agent paths as input, and camera-controlled generators such as ReCamDriving [39] use them to render scenes from new viewpoints. The physically plausible paths predicted by Hexa-P can serve as reliable inputs for these applications.

3 Preliminaries

Pedestrian Path Prediction

Given a scene with N agents, a path prediction network, $\mathcal{P}$, takes the past T_p frames as input and predicts the future path of a target agent over T_f frames. For each agent $i \in \{0, 1, \ldots, N-1\}$, where $i = 0$ for the target agent, the inputs to $\mathcal{P}$ are as follows:

- $\boldsymbol{\tau}^i_\mathrm{p} = (\boldsymbol{\tau}^i_{\mathrm{p},-T_\mathrm{p}+1}, \cdots, \boldsymbol{\tau}^i_{\mathrm{p},0}) \in \mathbb{R}^{T_\mathrm{p}\times 2}$: Past 2D path on the ground plane.
- $\boldsymbol{J}^i_\mathrm{p} = (\boldsymbol{J}^i_{\mathrm{p},-T_\mathrm{p}+1}, \cdots, \boldsymbol{J}^i_{\mathrm{p},0}) \in \mathbb{R}^{T_\mathrm{p}\times J\times 3}$: Past 3D positions of J joints.

$\mathcal{P}$ is trained with the following loss, $\mathcal{L}_\mathrm{f}$, to predict a target agent's 2D future path on the ground plane, $\hat{\boldsymbol{\tau}}_\mathrm{f} \in \mathbb{R}^{T_\mathrm{f}\times 2}$, which is expected to match its ground-truth, $\boldsymbol{\tau}_\mathrm{f}$:

$$\mathcal{L}_\mathrm{f} = \mathrm{MSE}(\hat{\boldsymbol{\tau}}_\mathrm{f}, \boldsymbol{\tau}_\mathrm{f}) \tag{1}$$

Locomotion Generation

Humanoid locomotion can be generated to follow any given path, $\boldsymbol{\tau}_\mathrm{s} \in \mathbb{R}^{T_\mathrm{f}\times 2}$, in a physics simulator. A locomotion generator, $\mathcal{G}$, takes $\boldsymbol{\tau}_\mathrm{s}$ and the current humanoid state, $\boldsymbol{h}_0$, to control the humanoid joints, where $\boldsymbol{h}_0$ comprises their positions, rotations, velocities, and angular velocities.

By reinforcement learning, $\mathcal{G}$ is trained by maximizing the discounted cumulative reward, $\Omega \in [0, 1]$ [1]. While it is difficult to control the humanoid perfectly to follow $\boldsymbol{\tau}_\mathrm{s}$ due to kinematic constraints such as joint-angle and torque limits, $\mathcal{G}$ is trained such that a path executed in a physics simulator (denoted by $\boldsymbol{\tau}_\mathrm{e}$) becomes as close to $\boldsymbol{\tau}_\mathrm{s}$ as possible. As $\boldsymbol{\tau}_\mathrm{e}$ gets closer to $\boldsymbol{\tau}_\mathrm{s}$, Ω becomes larger. We therefore use Ω as a measure of physical plausibility: low values indicate paths that are infeasible under physical constraints; conversely, high values indicate physically achievable paths.

Path Prediction using Generated Locomotion

For training $\mathcal{P}$ with the aforementioned generated locomotion data, the loss (Eq. (1)) is rewritten as follows:

$$\mathcal{L}_\mathrm{e} = \mathrm{MSE}(\hat{\boldsymbol{\tau}}_\mathrm{f}, \boldsymbol{\tau}_\mathrm{e}) \tag{2}$$

4 Proposed Method

The key idea of Hexa-P is to decompose the overall learning problem into physical-constraint modeling and social-interaction modeling within a three-stage framework. Figure 2 illustrates the overview of Hexa-P, which consists of three stages: two training stages and an inference stage. In Stage 1 (Sec. 4.1), physics simulation is used to pre-train the Physical Core jointly with a path predictor and a plausibility evaluator, which are "Path-Head" and "LocoVal-Head" in Fig. 2 (a), respectively. In Stage 2 (Sec. 4.2), Hexa-P fine-tunes the "Physical Core" with adapters and "Social module with TaPE" to preserve physical knowledge of the pose-path relationship while learning social path interactions in multi-agent scenarios, as shown in Fig. 2 (b). In Stage 3 (Sec. 4.3), dual-mode inference enables filtering of physically implausible path predictions (Fig. 2 (c)).

4.1 Stage 1: Physical Plausibility-aware Pre-training

4.1.1 Locomotion Generation

As shown in the upper part of Fig. 2 (a), we first generate diverse locomotion data using the locomotion generator, $\mathcal{G}$. Following [31], $\mathcal{G}$ is implemented as a policy network trained via reinforcement learning [40] using Isaac Gym [41] as a physics simulator. The discounted cumulative reward, Ω, is computed from a reward function that consists of a path-following reward, a motion style reward [30], and an energy penalty, as proposed in [31].

Each target path, $\boldsymbol{\tau}_\mathrm{s}$, for the humanoid is generated by randomly sampling velocities and turn angles with the following constraints used in PACER [31]: velocity in $[0.0005, 5]$ m/s, acceleration in $[0, 2]$ m/s^2, and turn rate in $[-180, 180]$ deg/s. For each training episode, we collect tuples $(\boldsymbol{\tau}^0_\mathrm{p}, \boldsymbol{\tau}_\mathrm{s}, \boldsymbol{J}^0_\mathrm{p})$ as inputs and $(\boldsymbol{\tau}_\mathrm{e}, \Omega)$ as

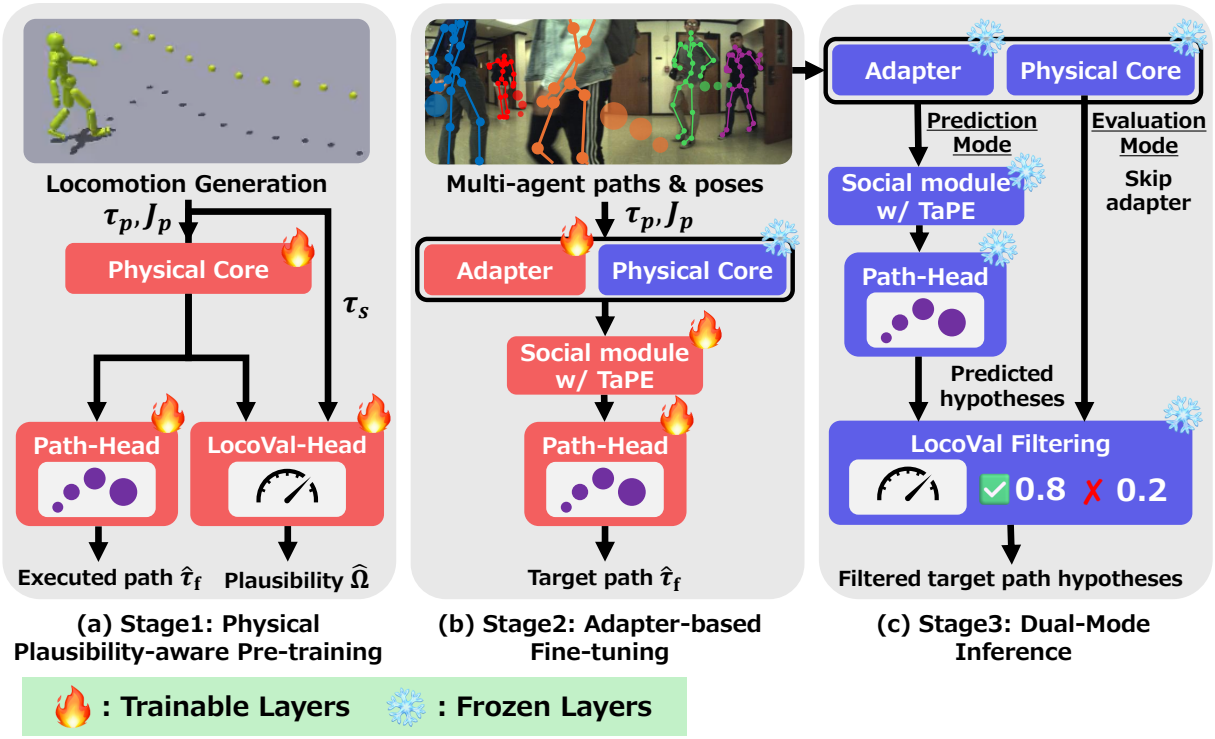


Fig. 2: Overview of Hexa-P. **(a) Pre-training:** In IsaacGym, a humanoid controller (PACER) tracks random target paths. We jointly train path prediction and plausibility estimation on *physically executed* motions, learning physical pose-path relationships. **(b) Fine-tuning:** The pre-trained Physical Core (blue) is frozen; only adapters (red) are trained. TaPE centers each agent to their own frame and encodes relative positions for multi-agent scenarios. **(c) Inference:** Dual-branch feature extraction enables plausibility-aware filtering by bypassing adapters to preserve the pre-trained plausibility estimation capability

ground-truth, where $\boldsymbol{J}_{\mathrm{p}}^{0}$ is sampled from a motion capture dataset [42].

4.1.2 Joint Pre-training of Path Prediction and Plausibility Estimation

As shown in Fig. 2 (a), the Physical Core, $\mathcal{C}$, serves as a shared feature extractor for two prediction heads that are jointly pre-trained:

- **Path-Head** predicts M path hypotheses for stochastic prediction from the Physical Core features. M independently initialized linear layers are used as the Path-Head to produce diverse path hypotheses.
- **LocoVal-Head** takes the target path, $\boldsymbol{\tau}_{\mathrm{s}}$, together with the Physical Core features and estimates a normalized plausibility score, $\hat{\Omega} \in [0, 1]$, via a sigmoid activation.

This joint pre-training in physics simulation provides three key benefits: (1) Pre-training only in physics simulation allows us to collect a large amount of training data. (2) The Physical Core learns to represent pose-path relationships for path prediction in the Path-Head while explicitly measuring their physical plausibility in the LocoVal-Head. (3) The LocoVal-Head can also serve as a plausibility evaluator at inference to filter out implausible path hypotheses, as proposed in [1]. The positive effects of these three benefits are empirically validated in Sec. 5.5.

The pre-training loss combines path prediction and plausibility estimation as follows:

$$\mathcal{L}_{\mathrm{pre}} = \lambda_{\mathrm{path}}\mathcal{L}_{\mathrm{path}} + \lambda_{\mathrm{locoval}}\mathcal{L}_{\mathrm{locoval}} \qquad (3)$$

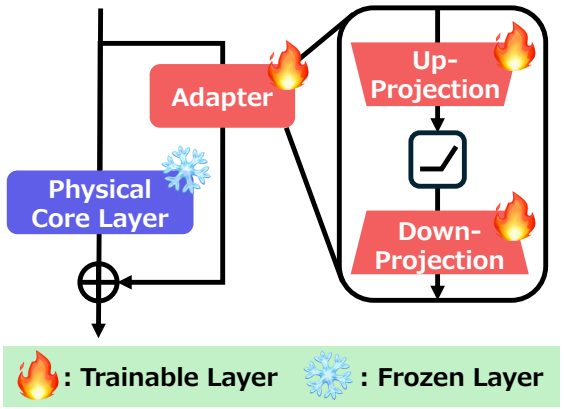


Fig. 3: Adapter-based fine-tuning architecture. Trainable bottleneck adapters are inserted in parallel with each layer of the frozen Physical Core. Skip connections enable bypassing adapters for plausibility estimation using pre-trained weights

$$\mathcal{L}_{\text{path}} = \mathbf{1}\{\text{No fall}\} \cdot \Omega \cdot \min_{m \in \{1,\dots,M\}} (\text{MSE}(\hat{\boldsymbol{\tau}}_{\text{f}}^{(m)}, \boldsymbol{\tau}_{\text{e}})) \quad (4)$$

$$\mathcal{L}_{\text{locoval}} = \text{MSE}(\hat{\Omega}, \Omega) \quad (5)$$

In Eq. (4), the path predictor $\mathcal{P}$ generates M stochastic predictions, where m indexes individual predictions. The minimum MSE among all M predictions is used to supervise the best prediction and promote diverse predictions. Ω is used as the weighting factor to encourage more plausible predictions. The indicator function, $\mathbf{1}\{\text{No fall}\}$, excludes failure cases where the humanoid falls during the training episode.

4.2 Stage 2: Adapter-Based Fine-tuning

Relying solely on the physical plausibility-aware pre-training described in Sec. 4.1, the Physical Core does not capture domain-specific pedestrian motions with real-world social interactions. Therefore, this Stage 2 (Fig. 2 (b)) addresses two challenges: (1) adapting the Physical Core to each dataset domain, while preserving physical knowledge of single-agent locomotion, using adapter-based fine-tuning (Sec. 4.2.1) and (2) bridging the gap between single-agent settings and socially-interactive real-world scenes with TaPE (Sec. 4.2.2).

4.2.1 Adapter Architecture

To learn dataset-specific features, in Stage 2: Adapter-based Fine-tuning of Hexa-P (Fig. 2 (b)), bottleneck adapters [43] are inserted in parallel with each layer of the pre-trained Physical Core, as shown in Fig. 3. While there are some alternative adaptation approaches such as LoRA [44] or in-context learning [45], we choose the adapters for their explicit skip-connection structure to achieve dual-mode feature extraction: by switching whether features propagate through the adapters ("Trainable Layers" in Fig. 3), the Physical Core can operate either with adapted features for path prediction or with original pre-trained features (from "Frozen Layers" in Fig. 3) for plausibility estimation.

Hexa-P freezes the pre-trained Physical Core and trains only the adapters for the following two reasons:

- **Physical knowledge preservation:** The frozen Physical Core retains the physical pose-path relationships, enabling pose-based prediction in short-observation scenarios where path cues alone are insufficient.
- **Efficient adaptation:** The adapters learn dataset-specific motion patterns with minimal parameter overhead, which does not compromise real-time deployment.

Within Physical Core, each Physical Core Layer (shown in Fig. 3) and the inserted Adapter Layer take the same hidden representation, $\boldsymbol{z} \in \mathbb{R}^d$, as an input:

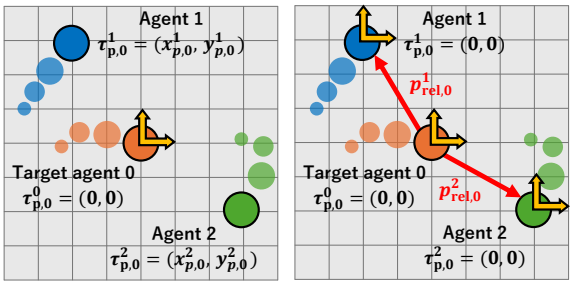
$$\boldsymbol{y} = \text{Physical Core Layer}(\boldsymbol{z}) + \text{Adapter Layer}(\boldsymbol{z}), \quad (6)$$

$$\text{Adapter Layer}(\boldsymbol{z}) = \boldsymbol{W}_{\text{up}} \cdot \phi(\boldsymbol{W}_{\text{down}} \cdot \boldsymbol{z}), \quad (7)$$

where $\boldsymbol{W}_{\text{down}} \in \mathbb{R}^{r \times d}$ and $\boldsymbol{W}_{\text{up}} \in \mathbb{R}^{d \times r}$ are learnable up- and down-projection layers, ϕ is a nonlinear activation, and r is the bottleneck dimension.

4.2.2 Target-relative Positional Encoding (TaPE)

A challenge in adapting the pre-trained model to multi-agent scenarios is the coordinate representation gap. During pre-training in Stage 1,



(a) Conventional Approach (b) Proposed Method (TaPE)

Fig. 4: Comparison of coordinate representations for multi-agent path prediction. (a) A conventional approach [5–8, 11] directly inputs all agent positions relative to the target agent. (b) Our TaPE first centers each agent to a target-centric coordinate (origin at current frame) for feature extraction, then encodes relative positions $\boldsymbol{p}^i_{\text{rel}}$ from the target agent after feature extraction

the Physical Core learns agent-centric representations in which the humanoid's future path always starts from the origin, *i.e.*, $(0, 0)$, which is identical to the humanoid's position $\boldsymbol{\tau}^0_{\text{p},0}$ in the last past frame (*i.e.*, the current frame, $t = 0$). However, as shown in Fig. 4 (a), conventional path prediction methods [5–7, 11] directly take all agents' coordinates relative to the target. For example, in Fig. 4 (a), all Target agent 0, agent 1, and agent 2 are represented in the single coordinate system whose origin is $\boldsymbol{\tau}^0_{\text{p},0}$. TaPE (Fig. 4 (b)) addresses this "single-agent to multi-agent" gap through two mechanisms: (1) agent-centric centering that preserves compatibility with the pre-training in Stage 1 and (2) relative position encoding that introduces the spatial distribution among all agents.

Agent-Centric Centering

Following the same coordinate transformation as pre-training, we center all inputs for each agent i independently, as shown in Fig. 4 (b):

$$\tilde{\boldsymbol{\tau}}^i_{\text{p},t} = \boldsymbol{\tau}^i_{\text{p},t} - \boldsymbol{\tau}^i_{\text{p},0} \tag{8}$$

$$\tilde{\boldsymbol{J}}^i_{\text{p},t,k} = \boldsymbol{J}^i_{\text{p},t,k} - \boldsymbol{J}^i_{\text{p},t,\text{pelvis}}, \quad k \in \{1, \ldots, J\} \tag{9}$$

Equation (8) centers each agent's path at their current position, ensuring compatibility with the path distribution learned in the pre-training. Equation (9) centers joint positions relative to the pelvis, removing global position information to focus on the body pose. $\tilde{\boldsymbol{\tau}}^i_{\text{p}}$ and $\tilde{\boldsymbol{J}}^i_{\text{p},k}$ are fed into the Physical Core to extract agent-centric features, $\boldsymbol{f}^i = \mathcal{C}(\tilde{\boldsymbol{\tau}}^i_{\text{p}}, \tilde{\boldsymbol{J}}^i_{\text{p}}) \in \mathbb{R}^d$.

Relative Position Encoding

After the aforementioned agent-centric feature extraction, we encode the inter-agent spatial distribution. For each agent i, we compute its relative position from the target agent 0:

$$\boldsymbol{p}^i_{\text{rel},t} = \boldsymbol{\tau}^i_{\text{p},t} - \boldsymbol{\tau}^0_{\text{p},t}. \tag{10}$$

$\boldsymbol{p}^i_{\text{rel},t}$ is processed through an MLP to produce a positional encoding, $\boldsymbol{e}^i_{\text{rel}} \in \mathbb{R}^d$. To handle missing observations in which either agent i or agent 0 is not visible, a binary visibility mask is also encoded and concatenated to the input features.

Social Module

Before the "Social module w/ TaPE" block in Fig. 2 (b), the positional encodings, $\boldsymbol{e}^i_{\text{rel}}$, are concatenated with agent-centric features, $\boldsymbol{f}^i$, extracted by the Physical Core, along with ID embeddings that distinguish the target agent from neighboring agents. These concatenated features are processed through the social module that aggregates information across all agents to model social interactions. Finally, the Path-Head predicts the target path $\hat{\boldsymbol{\tau}}_{\text{f}}$ as shown in Fig. 2 (b).

4.2.3 Fine-tuning Objective

The fine-tuning objective, $\mathcal{L}_{\text{rpath}}$, is as follows:

$$\mathcal{L}_{\text{rpath}} = \min_{m \in \{1,\ldots,M\}} (\text{MSE}(\hat{\boldsymbol{\tau}}^{(m)}_{\text{f}}, \boldsymbol{\tau}_{\text{f}})) \tag{11}$$

Unlike the pre-training loss in Eq. (4), the simulation-specific weights (*i.e.*, $\mathbf{1}\{\text{No fall}\}$ and Ω) are not used in $\mathcal{L}_{\text{rpath}}$ because real pedestrian datasets contain only non-falling paths and do not provide simulation-based plausibility scores.

4.3 Stage 3: Dual-mode Inference

Dual Modes

At inference time, the adapter architecture enables two operational modes, as illustrated in Fig. 2 (c). In the **Prediction Mode**, features pass through both frozen Physical Core layers and

adapters, combining physical priors with domain-specific knowledge learned during fine-tuning for real-world path prediction. In the **Evaluation Mode**, adapters ("Trainable Layers" in Fig. 3) are bypassed entirely. Since adapters are not trained for plausibility estimation during fine-tuning, this bypass prevents fine-tuned features from interfering with LocoVal-Head, allowing it to rely solely on physical knowledge learned during pre-training. As mentioned in Sec. 4.2.1, the evaluation mode preserves the reliability of single-agent physics plausibility estimation. We leverage this Evaluation Mode to filter implausible predictions, as described below.

LocoVal Filtering

Following EmLoco [1], we apply LocoVal filtering as an auxiliary post-processing for stochastic path prediction methods that generate multiple hypotheses. Using the Evaluation Mode described above, multiple path hypotheses generated at inference are evaluated to discard those with plausibility scores below the threshold θ. If all hypotheses fall below the threshold, the highest-scored hypothesis is retained.

This mechanism is designed to remove obviously implausible predictions (*e.g.*, abrupt direction changes or unrealistic accelerations) based on single-agent locomotion learned in simulation. In Sec. 5.9, we show that this improves downstream navigation safety.

Such filtering is necessary because fine-tuning on noisy real-world data can degrade path plausibility, especially in short observation scenarios where single-agent dynamics dominate. To address this, Hexa-P employs the LocoVal-Head trained on simulated pose-path data. The filtering threshold θ can be adjusted based on application requirements, trading off between coverage and the strictness of filtering.

5 Experiments

We evaluate Hexa-P across various urban navigation scenarios for unmanned systems. First, we describe our experimental setup (Sec. 5.1) and compare Hexa-P against baseline methods under sufficient observation (Sec. 5.2) and short-observation settings (Sec. 5.3). While our primary focus is stochastic prediction, we also evaluate deterministic prediction performance against state-of-the-art methods (Sec. 5.4). We then validate each component through ablation studies (Sec. 5.5) and assess physical plausibility (Sec. 5.6). We also verify that the predicted paths are dynamically executable by a physical humanoid in simulation (Sec. 5.7) and evaluate LocoVal filtering (Sec. 5.8). Furthermore, we measure the downstream navigation-safety benefit when our predictions are supplied to a planner (Sec. 5.9), report the inference cost (Sec. 5.10), and analyze failure cases under the flat-ground prior (Sec. 5.11).

5.1 Experimental Setup

5.1.1 Datasets

Pre-training

In this stage, Hexa-P is pre-trained using the pre-trained locomotion generator $\mathcal{G}$. Locomotion sequences are generated using PACER [31] in IsaacGym [41] by running 4,096 environments in parallel at 5 fps. Following PACER [31], we use 285 locomotion sequences from the AMASS dataset [42] as past motion sequences during pre-training. Each target path, generated by the path generator described in Sec. 4.1.1, is then aligned with the humanoid state at $t = 0$ following EmLoco [1]: $\boldsymbol{\tau}_{\mathrm{s}}$ is rotated and scaled so that its heading vector matches the humanoid's heading vector at $t = 0$.

Fine-tuning and Inference

We evaluate Hexa-P on three datasets that represent different aspects of traffic scenes:

- **JackRabbot Dataset and Benchmark (JRDB)** [46]: A real-world dataset collected by a navigating robot, representing an unmanned system navigation scenario in open urban environments. Out of 27 sequences with available annotations, we use 10 for training, 1 for validation, and 2 for testing. As this dataset lacks 3D pose annotations, we use SMPL poses (24 joints) [47] estimated by [27].
- **Joint Track Auto (JTA)** [48]: A synthetic dataset with 3D pose annotations (22 joints) comprising 256/128/128 sequences for training/validation/testing.
- **Pedestrians and Cyclists in Road Traffic (Urban)** [2]: Real traffic scenario dataset

directly relevant to autonomous driving applications, containing more than 2000 paths of pedestrians and cyclists with 3D poses (13 joints). We focus exclusively on pedestrians as Hexa-P features human locomotion.

For JRDB and JTA, we use the same data split as Social-Trans [7] and predict 12 frames (4.8 s) from 9 observed frames (3.2 s) at 2.5 fps. For Urban, we use the official split and predict 12 frames (2.4 s) from 4 frames (0.8 s) at 5 fps.

5.1.2 Metrics

Predictions are evaluated with ADE/FDE (Average/Final Displacement Error) and their best-of-M variants (minADE/minFDE) following standard practice [5–7, 11]. We also report χ^2 distances [49] measuring physical plausibility in terms of velocity, acceleration, angular velocity, and angular acceleration compared to ground truth.

5.1.3 Baselines

We compare Hexa-P against Social-Trans [7] (our base architecture, modified with bi-directional encoding and dynamic masking [28] for fair comparison) and EmLoco [1] (using the same modified architecture). Results for other methods (Vanilla-LSTM [17], Social-LSTM [5], Social-GAN [6], Trajectron++ [11], Dir-LSTM/Dir-social-LSTM [3], Transformer [22], AutoBots [50], EqMotion [8]) are cited from [4, 7].

5.1.4 Implementation Details

Network Architecture

We build upon Social-Trans [7] as our base architecture. The Physical Core consists of 6 Transformer layers with $d = 128$, 4 attention heads, and feed-forward dimension of 1024. A bi-directional encoder [28] assigns learnable positional embeddings to each frame based on its time offset from the current frame ($t = 0$), separately for past and future directions. By conditioning on elapsed time rather than frame index, the same positional encoding applies consistently across different frame rates, enabling the Physical Core pre-trained at 5 fps to operate on fine-tuning datasets at different frame rates (*e.g.*, 2.5 fps for JRDB and JTA) without modification. Attention masking lets the Physical Core handle variable joint configurations and observation lengths, supporting adaptation to datasets with different pose configurations. The LocoVal-Head is a 3-layer MLP with validity encoding for varying frame rates. The Path-Head uses $M = 5$ linear projections for stochastic prediction ($M = 1$ for deterministic).

After pre-training, bottleneck adapters with dimension r are inserted in parallel to both multi-head attention and feed-forward layers. We set $r = 32$ on JRDB and JTA, and $r = 16$ on Urban. We use a smaller adapter dimension on Urban due to its shorter observation length and smaller dataset size. The social module consists of 3 Transformer layers with the same hidden dimension $d = 128$ and 4 attention heads as the Physical Core, aggregating features across agents using learnable ID embeddings and relative position encodings from TaPE (4-layer MLP with validity encoding).

Pre-training

We train the Physical Core jointly with Path-Head and LocoVal-Head using 4,096 parallel environments for 2,000 episodes, each running 145 timesteps at 30 Hz with a physics timestep of 1/60 s, resulting in approximately 1.19 billion environment steps in total.

This pre-training is a one-time offline cost, requiring 21.7 h on a single NVIDIA A100 GPU. None of this cost is incurred at deployment, where the model is lightweight and runs in real time as shown in Sec. 5.10.

Path prediction is performed at 5 fps ($T_p = 17$, $T_f = 24$). Owing to the bi-directional encoder [28], this frame rate is not fixed after pre-training; the Physical Core can generalize to other frame rates during fine-tuning, as positional encodings are assigned based on elapsed time from the current frame rather than frame index. We use AdamW optimizer (lr 5×10^{-4}, weight decay 10^{-4}) with cosine annealing. Loss weights $\lambda_{\text{path}} = \lambda_{\text{locoval}} = 0.5$ are balanced via uncertainty weighting [51] and gradient surgery is applied [52].

To improve robustness, we apply random frame dropout (rate 0.5) and dynamic spatial-temporal masking [28] with joint dropout ratio $\rho_{\text{joint}} \sim \text{Uniform}[0, 1]$.

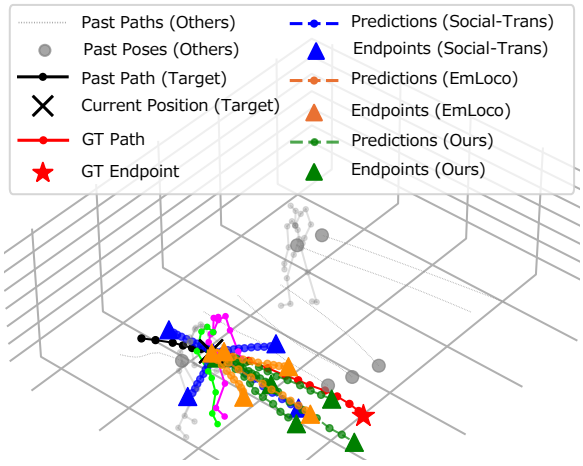


Fig. 5: Path prediction results with sufficient observations on JRDB dataset ($M = 5$, $T_{\mathrm{p}} = 9$). Poses are visualized in magenta and lime green

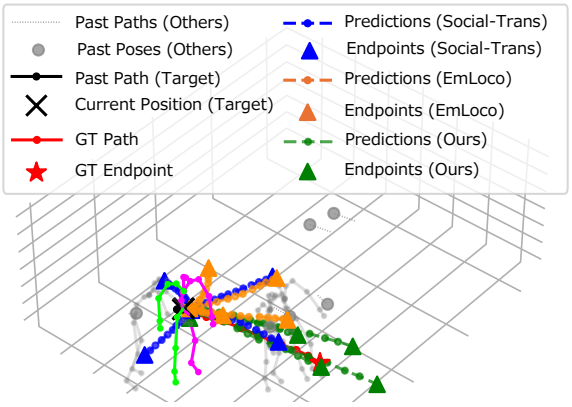


Fig. 6: Path prediction results with short observations on JRDB dataset ($M = 5$, $T_{\mathrm{p}} = 2$). Poses are visualized in magenta and lime green

Fine-tuning

We freeze the pre-trained Physical Core and LocoVal-Head, training only the adapters (∼0.1M parameters, ∼3% of total), Social module, and Path-Head using AdamW (lr $2{\times}10^{-4}$, weight decay 10^{-4}). We train for 100 epochs on JRDB, and 50 epochs on JTA and Urban. We apply modality masking (30%) and frame/joint-level dynamic spatial-temporal masking to simulate partial observations. For JTA and Urban, we mask out joints not common to SMPL [47] for compatibility with pre-training. Since JRDB and JTA are collected at 2.5 fps, which differs from the pre-training frame rate of 5 fps, the bi-directional encoder bridges this gap by assigning time-relative positional encodings to each frame, requiring no modification to the frozen Physical Core.

LocoVal filtering

The filtering threshold, θ, is determined by selecting the value that minimizes ADE on the validation set. Specifically, we evaluate 101 uniformly spaced threshold candidates in $[0, 1]$ and choose the one yielding the lowest ADE for retained predictions.

5.2 Results with Sufficient Observations

Table 1 presents stochastic prediction results with sufficient observations ($T_{\mathrm{p}} = 9$).

On JRDB, while EmLoco improves over Social-Trans (ADE: 0.56 vs. 0.58) through its auxiliary EmLoco loss, Hexa-P achieves further improvement (ADE: 0.51). This result demonstrates the advantage of direct pre-training over the two-step learning strategy of EmLoco. Both minADE/minFDE are improved (0.22/0.39 vs. 0.24/0.43 for both Social-Trans and EmLoco). On JTA, Hexa-P reduces ADE/FDE (1.49/3.00) compared to EmLoco (1.51/3.18) while maintaining comparable minADE/minFDE.

Figure 5 provides a qualitative comparison on the JRDB dataset. Despite incorporating 3D poses, predictions of Social-Trans are scattered widely, failing to leverage pose information to constrain physically implausible trajectories. EmLoco shows more plausible predictions through its auxiliary EmLoco loss, but its predictions remain distant from the GT path, suggesting that its ability to exploit pose-path relationships is limited for accurate prediction. In contrast, Hexa-P produces predictions that are both physically consistent with the observed poses and spatially close to the GT path.

5.3 Robustness to Short Observations

Table 2 evaluates robustness when only two frames ($T_{\mathrm{p}} = 2$) are observed, simulating safety-critical scenarios where pedestrians suddenly appear.

On JRDB, the difference between Hexa-P and EmLoco becomes substantial. While EmLoco

Table 1: Prediction Performance with Sufficient Observation ($M = 5$, $T_p = 9$). **Best** and **second best** results are highlighted

Dataset	Method	ADE↓	FDE↓	minADE↓	minFDE↓
JRDB	Social-Trans	0.58	1.20	**0.24**	**0.43**
	EmLoco	**0.56**	**1.12**	**0.24**	**0.43**
	Hexa-P (Ours)	**0.51**	**0.99**	**0.22**	**0.39**
JTA	Social-Trans	1.55	3.27	**0.57**	**1.01**
	EmLoco	**1.51**	**3.18**	**0.57**	**1.01**
	Hexa-P (Ours)	**1.49**	**3.00**	**0.60**	**1.04**

Table 2: Prediction Performance with Short Observation ($M = 5$, $T_p = 2$). **Best** and **second best** results are highlighted

Dataset	Method	ADE↓	FDE↓	minADE↓	minFDE↓
JRDB	Social-Trans	1.29	2.43	0.26	0.46
	EmLoco	**1.13**	**2.01**	**0.25**	**0.44**
	Hexa-P (Ours)	**0.64**	**1.23**	**0.23**	**0.40**
JTA	Social-Trans	1.81	3.90	**0.68**	**1.22**
	EmLoco	**1.80**	**3.87**	**0.68**	**1.22**
	Hexa-P (Ours)	**1.62**	**3.35**	**0.66**	**1.18**

improves over Social-Trans (ADE: 1.13 vs. 1.29), Hexa-P achieves 0.64, representing a **43% reduction over EmLoco** and **50% over Social-Trans**. This gap suggests that the Physical Core provides robust pose-path representations even when observations are limited. On JTA, EmLoco shows minimal improvement over Social-Trans (1.80 vs. 1.81), while Hexa-P maintains a clear gain (1.62). Compared with Table 1, the degradation from sufficient to short observations is smallest for Hexa-P (1.49→1.62) compared to both EmLoco (1.51→1.80) and Social-Trans (1.55→1.81).

Figure 6 provides qualitative comparison. Compared to the sufficient observation case (Fig. 5), the performance gap between methods becomes more pronounced. Social-Trans produces highly scattered predictions, indicating severe degradation when temporal information is limited. EmLoco shows improved predictions over Social-Trans, but still generates physically implausible paths that deviate significantly from the GT path, suggesting that its auxiliary loss-based approach is insufficient to promote physical constraints. In contrast, Hexa-P maintains prediction quality similar to the sufficient observation case, producing paths that remain both physically plausible and close to the GT path.

This robustness strongly supports our hypothesis that our pre-trained Physical Core provides stable pose-path representations even when past observations are severely limited. The comparison with EmLoco further demonstrates that direct pre-training on physics simulation is more effective than using simulation indirectly for auxiliary loss computation (as illustrated in Fig. 1 (b) vs. (c)).

5.4 Benchmarking with Deterministic Setting

Table 3 compares Hexa-P with state-of-the-art methods under the deterministic setting ($M = 1$) across three datasets.

On JRDB, Hexa-P achieves the best results (0.36/0.68), outperforming all baselines, including pose-based methods (indicated by †). While

Table 3: Deterministic prediction performance ($M = 1$). JRDB and JTA use $T_p = 9$ at 2.5 fps; Urban uses $T_p = 4$ at 5 fps. † indicates methods using 3D pose as input. **Best** and **second best** results are highlighted

Method	JRDB		JTA		Urban	
	ADE↓	FDE↓	ADE↓	FDE↓	ADE↓	FDE↓
Path-only methods						
Vanilla-LSTM [17]	0.42	0.83	1.44	3.25	0.58	1.06
Social-LSTM [5]	0.47	0.95	1.21	2.54	0.58	1.06
Social-GAN [6]	0.50	0.99	1.66	3.76	0.60	1.08
Trajectron++ [11]	0.40	0.78	1.18	2.53	-	-
Dir-LSTM [3]	0.45	0.87	1.37	3.06	0.58	1.06
Dir-social-LSTM [3]	0.48	0.95	1.23	2.59	0.58	1.06
Transformer [22]	0.56	1.10	1.56	3.54	0.60	1.11
AutoBots [50]	0.39	0.80	1.20	2.70	0.58	1.04
EqMotion [8]	0.42	0.78	1.13	2.39	0.58	1.05
Path and Pose-based methods						
Social-Trans† [7]	0.38	0.73	**0.97**	2.00	**0.57**	1.03
EmLoco† [1]	**0.37**	**0.70**	1.00	**1.92**	**0.57**	**1.02**
Hexa-P (Ours)†	**0.36**	**0.68**	**0.99**	**1.98**	**0.56**	**1.01**

the margin is smaller than in stochastic settings (Table 1), Hexa-P consistently outperforms Social-Trans (0.38/0.73) and EmLoco (0.37/0.70), demonstrating that physical plausibility-aware pre-training benefits even deterministic prediction. On JTA, Hexa-P (0.99/1.98) performs comparably to Social-Trans (0.97/2.00) and EmLoco (1.00/1.92). The synthetic nature of JTA with abundant and accurate pose annotations may reduce the relative advantage of physics-based pre-training. On Urban [2], which contains real traffic scenes, Hexa-P also outperforms all baselines. In particular, although the Urban dataset features shorter observation (0.8 s) and prediction (2.4 s) horizons and a different frame rate (5 fps) than JRDB/JTA, Hexa-P achieves the best performance, further demonstrating that physical knowledge learned in simulation transfers effectively to real traffic environments.

5.5 Ablation Studies

Table 4 validates each component of Hexa-P on JRDB.

Pre-training effectiveness

To evaluate the effectiveness of physics-informed pre-training, we first examine performance within the physics simulator without any fine-tuning. Notably, (b) pre-training alone already shows strong improvements over (a) the randomly initialized predictor (ADE: 0.79 vs. 1.27 for sufficient observations), demonstrating that Hexa-P learns pose-path representations from simulation data.

Pre-training ablation

We next examine how variations in pre-training configuration affect final performance after fine-tuning. (c) Ablating the Path Head from pre-training (training only the LocoVal Head) significantly degrades performance (ADE: 1.06/0.97), confirming the importance of joint training. (d) Reducing pre-training from 2000 to 500 training episodes degrades performance (ADE: 0.69/0.78 vs. 0.51/0.64 for $T_p = 9/2$). However, even with this reduced pre-training resource, Hexa-P still outperforms Social-Trans and EmLoco in short-observation scenarios (0.78 vs. 1.29/1.13), demonstrating the importance of learning physics-based pose-path relationships.

Table 4: Ablation and component analysis on JRDB ($M = 5, T_{\mathrm{p}} = 9$ and 2). **Best** and **second best** results are highlighted

Method	$T_{\mathrm{p}} = 9$		$T_{\mathrm{p}} = 2$	
	ADE↓	minADE↓	ADE↓	minADE↓
Social-Trans [7]	0.58	0.24	1.29	0.26
EmLoco [1]	0.56	0.24	1.13	0.25
Pre-training ablation				
(a) Initialized	1.27	1.01	1.28	0.95
(b) w/o Fine-tuning	0.79	0.38	0.88	0.49
(c) w/o Path Pre-training	1.06	0.25	0.97	0.25
(d) w/ Pre-training (500 episodes)	0.69	0.24	0.78	**0.24**
Fine-tuning ablation				
(e) w/o Social module	0.77	0.26	0.83	0.26
(f) w/o Adapter	0.66	0.24	0.75	0.25
(g) w/o TaPE	0.56	0.24	0.89	**0.24**
(h) w/ LoRA	0.66	0.24	0.74	0.26
(i) w/ FFT	**0.54**	**0.22**	0.72	**0.23**
(j) w/ Adapter ($r = 8$)	0.55	**0.23**	**0.64**	**0.23**
(k) w/ Adapter ($r = 16$)	0.55	**0.23**	**0.66**	**0.24**
Hexa-P (Full, $r = 32$)	**0.51**	**0.22**	**0.64**	**0.23**

Fine-tuning components

(e) Removing the social module degrades performance in both settings, confirming the importance of social interactions. (f) Without adapter-based fine-tuning, training only the Social Module and Path Head while freezing the Physical Core results in degraded performance (ADE: 0.66/0.75), as the frozen core cannot adapt to dataset-specific behaviors. (g) Without TaPE, directly inputting relative positions to the model results in performance degradation especially in short-observation scenarios (ADE: 0.89 for $T_{\mathrm{p}} = 2$), validating its role in bridging single-agent pre-training to multi-agent scenarios. (h), (i) We compare two alternative fine-tuning strategies to our adapter-based approach. Using LoRA [44] instead of adapters results in limited representational capacity due to its lack of nonlinear transformations (ADE: 0.66/0.74). Full fine-tuning (FFT), which updates all model parameters, achieves the second-best accuracy with sufficient observation (ADE: 0.54) but suffers from performance degradation in short-observation scenarios (ADE: 0.72) as it overwrites pre-trained physical knowledge. Our adapter-based method effectively balances adaptation capacity with knowledge preservation. (j), (k) Adapters with smaller bottleneck dimensions ($r = 8, 16$) use fewer parameters while maintaining comparable performance, demonstrating the parameter efficiency of our adapter-based approach.

Novelty over EmLoco and contribution decomposition

We further use Table 4 to identify where Hexa-P's improvement comes from over EmLoco [1]. Both the Social-Trans and EmLoco baselines use the same modified backbone as Hexa-P, so the differences reported below are not caused by the backbone.

The main distinction from EmLoco is joint pre-training rather than the auxiliary-loss scheme. When we ablate (c) the Path-Head from pre-training and learn plausibility alone, ADE degrades to 1.06/0.97, compared with 0.51/0.64 for the full model. This confirms that learning path prediction and plausibility jointly in simulation is the main cause of the improvement.

The adapters and TaPE provide additional gains. (f) Freezing the Physical Core without

adapters degrades ADE to 0.66/0.75, and both (h) LoRA (0.66/0.74) and (i) full fine-tuning (0.54/0.72) are worse in the safety-critical short-observation setting. Adapter-based fine-tuning thus balances adaptation and preservation of the pre-trained physical knowledge better than these alternatives. (g) Removing TaPE degrades ADE to 0.89 at $T_{\mathrm{p}} = 2$, confirming its role in transferring single-agent pre-training to multi-agent scenes. Overall, Hexa-P's improvement comes primarily from joint pre-training, with the adapters and TaPE contributing additional gains.

5.6 Physical Plausibility Assessment

In stochastic prediction, all predicted paths must remain physically plausible. Even if one prediction matches the ground truth well, a single implausible path among the alternatives can mislead a downstream planner that must consider all hypotheses for safe operation. From this perspective, Table 5 evaluates physical plausibility using χ^2 distances on JRDB. Lower values indicate predictions closer to ground-truth motion statistics.

With sufficient observation ($T_{\mathrm{p}} = 9$), Hexa-P achieves the best or second-best plausibility across all metrics. Notably, velocity and acceleration χ^2 distances are substantially reduced (0.014/0.005 vs. 0.029/0.030 for Social-Trans), indicating smoother, more natural predictions. With short observation ($T_{\mathrm{p}} = 2$), the improvements are even larger. In particular, the velocity χ^2 distance drops from 0.493 (Social-Trans) to 0.018 (Hexa-P), demonstrating that baseline predictions exhibit unrealistic velocity patterns when observation is limited, while Hexa-P maintains physically plausible predictions.

These results confirm that Hexa-P generates not only accurate but also physically plausible paths. In Sec. 5.9, we further show that Hexa-P improves downstream navigation safety.

5.7 Path Executability in Simulation

The χ^2 distances in Sec. 5.6 show that the predicted motion statistics are close to those of real data. To further test that a predicted path can actually be executed by a physical humanoid, we feed the predicted paths into the physics simulator and check whether PACER [31] can follow them. Each path is rolled out for 145 control steps (about 4.8 s) over 2048 trials. We report the fall rate, the mean tracking error between the pelvis and the target path, and the success rate. A trial is successful if the humanoid does not fall and ends within 0.5 m of the goal. As a reference, we also roll out the sampled target paths generated by the path generator as described in **??**.

Table 6 shows that the paths predicted by Hexa-P are followed with a 0.2% fall rate, an 8.7 cm tracking error, and a 99.4% success rate. Notably, they are followed even more reliably than the sampled target paths (3.1% falls, 92.8% success), which are the reference distribution used to train the locomotion generator. These results indicate that Hexa-P produces physically executable paths.

5.8 LocoVal Filtering at Inference

Table 7 evaluates LocoVal filtering on JRDB. We compare the ADE of retained predictions ("Kept") versus filtered-out ones ("Rejected"), where higher Ratio values indicate more effective discrimination. Hexa-P with Evaluation Mode (bypassing adapters) achieves ratios of 2.57 and 3.33 for minADE at $T_{\mathrm{p}} = 9$ and $T_{\mathrm{p}} = 2$, respectively, substantially outperforming EmLoco [1] (0.76 and 0.82). Comparing Evaluation Mode with Prediction Mode (1.85 and 1.92 at $T_{\mathrm{p}} = 9$ and 2, respectively) also confirms that bypassing adapters preserves the pre-trained plausibility estimation capability.

5.9 Downstream Navigation Safety

To directly measure the navigation safety benefit, we place the predictor inside a downstream navigator and evaluate collision-risk. We use CrowdNav++ simulator [53], in which a robot navigates from a start to a goal among pedestrian trajectories replayed from the dataset. The navigator is a training-free Social-Force model [9] that treats each predicted point as an obstacle, adding its repulsive force to the model's own distance-based decay. We evaluate on the JTA replay (randomly sampled 100 episodes from test split), which contains dense and dynamic crowds that frequently require collision avoidance. The Social-Force strength A and the LocoVal threshold θ are tuned on the JTA *validation* split.

Table 8 shows that safety improves consistently as stronger predictions are used. Adding

Table 5: Physical Plausibility Evaluation ($M = 5$, $T_p = 9/2$). "Vel." and "Acc." denote velocity and acceleration, respectively. **Best** and **second best** results are highlighted

Method	χ^2 distances↓			
	Vel.	Acc.	Angular Vel.	Angular Acc.
Social-Trans	0.029/0.493	0.030/**0.015**	0.102/0.171	0.128/0.292
EmLoco	**0.027**/**0.171**	**0.025**/0.037	**0.032**/**0.076**	**0.014**/**0.073**
Hexa-P (Ours)	**0.014**/**0.018**	**0.005**/**0.002**	**0.030**/**0.033**	**0.026**/**0.009**

Table 6: Executability of predicted paths, rolled out by PACER in simulation. Success = no fall and final distance < 0.5 m. **Best** results are highlighted

Method	Fall rate↓	Tracking error (m)↓	Success rate↑
Sampled target path	3.1%	0.339	92.8%
Hexa-P (Ours)	**0.2%**	**0.087**	**99.4%**

any learned prediction reduces the collision rate from 35% without prediction to 14% with Social-Trans. Using Hexa-P further lowers collisions to 10%. Applying LocoVal filtering removes the remaining implausible predictions, reducing collisions to 8% and raising the success rate to 92%.

5.10 Inference Cost

Because path prediction is intended for real-time robotic and autonomous-driving systems, we report the inference cost at deployment.

Table 9 breaks down the inference cost of Hexa-P on a single RTX 6000 Ada GPU with batch size 1. Most of the cost comes from the base predictor (Social-Trans) at 5.56 ms. Adding TaPE increases the latency to 5.79 ms, and adding the adapters, which account for only 3.1% of the parameters, brings it to 6.54 ms. This corresponds to about 150 fps, which is real-time on a single GPU. LocoVal filtering adds a further 0.5 ms, and this cost is practically independent of the number of hypotheses M assuming sufficient memory, because all hypotheses can be processed in parallel. Plausibility-aware filtering therefore remains inexpensive even with many hypotheses.

5.11 Failure Cases under the Flat-Ground Prior

Hexa-P is trained with locomotion on a flat-plane generated by PACER, and it is therefore not reliable for pedestrians who are not walking. Figure 7 shows two representative cases from the JRDB test set, both involving near-stationary pedestrians with an estimated 3D pose. For the bending pedestrian in Fig. 7(a), the body leans forward while standing and faces the direction opposite to the past trajectory, so Hexa-P predicts in the wrong direction. For the person eating while seated in Fig. 7(b), the body is off the ground and therefore lies outside the walking distribution on which Hexa-P is trained. In both cases the ground-truth future path barely moves, but the prediction by Hexa-P includes spreading trajectories.

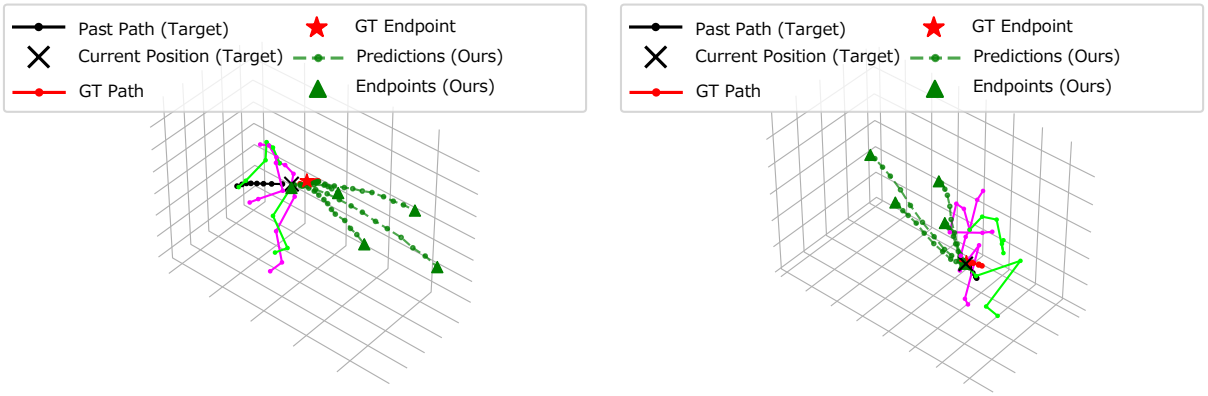
6 Conclusion

This paper presents Hexa-P, a framework that leverages physics simulation as a pre-training resource for pedestrian path prediction in urban environments. By jointly pre-training path prediction and plausibility estimation, then adapting to multi-agent scenes through adapter-based fine-tuning with TaPE, Hexa-P achieves state-of-the-art accuracy and physical plausibility across three datasets. In particular, Hexa-P reduces ADE by up to 50% in safety-critical short-observation scenarios on JRDB, demonstrating that physics-based pose-path priors transfer effectively to real-world data even with estimated 3D poses.

Despite these strengths, our approach has some limitations. First, the physics simulation assumes a flat ground plane as mentioned in Sec. 5.11. Second, while Hexa-P focuses on pedestrians, safety-critical traffic scenarios also involve

Table 7: LocoVal Filtering on JRDB ($M = 5$). Kept and Rejected: predictions accepted and filtered out. Ratio = Rejected/Kept; a higher ratio indicates more effective discrimination

Method		$T_p = 9$		$T_p = 2$	
		ADE	minADE	ADE	minADE
w/o Filtering		0.51	0.22	0.64	0.23
EmLoco	Kept	0.35	0.55	0.46	0.57
	Rejected	0.76	0.42	0.81	0.47
	Ratio↑	**2.17**	0.76	1.76	0.82
Hexa-P (Prediction Mode)	Kept	0.46	0.34	0.57	0.37
	Rejected	0.82	0.63	0.93	0.71
	Ratio↑	1.78	**1.85**	**1.63**	**1.92**
Hexa-P (Ours)	Kept	0.49	0.28	0.61	0.30
	Rejected	1.02	0.72	1.21	1.00
	Ratio↑	**2.08**	**2.57**	**1.98**	**3.33**



(a) Bending while standing (b) Eating while seated

Fig. 7: Failure cases on JRDB with an estimated 3D pose. The observed pose at current frame is visualized in magenta and lime green

Table 8: Downstream navigation safety on JTA replay. **Best** results are highlighted

Method	Collision↓	Success↑
None (w/o prediction)	35%	65%
Social-Trans [7]	14%	86%
Hexa-P w/o LocoVal	10%	90%
Hexa-P w/ LocoVal	**8%**	**92%**

Table 9: Inference-cost decomposition of Hexa-P on a single NVIDIA RTX 6000 Ada GPU (bs = 1)

Method (cumulative)	Params	Latency
Baseline (Social-Trans)	3.16 M	5.56 ms
+ TaPE	3.18 M	5.79 ms
+ Adapter (r=32) **(Ours)**	**3.28 M**	**6.54 ms**
+ LocoVal filtering	+1.1 k	+0.50 ms

cyclists and other vulnerable road users whose motion dynamics differ from pedestrian locomotion. Future directions include incorporating terrain-aware and activity-aware simulation to

handle diverse urban environments and non-walking behavior, extending the framework to other road users such as cyclists.

Author contributions. Hiromu Taketsugu: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing (original draft). Shohei Nobuhara: Methodology, Project administration, Supervision, Writing (review and editing). Norimichi Ukita: Funding acquisition, Methodology, Project administration, Resources, Supervision, Writing (review and editing).

Funding. This work was supported by JSPS KAKENHI Grant Number JP25KJ2195.

Data availability. The JRDB [46], JTA [48], Urban [2], and AMASS [42] datasets used in this study are publicly available from their sources. No new datasets are distributed with this work.

Code availability. The implementation of Hexa-P is available at https://anonymous.4open.science/r/Hexa-P_PAA-6CEC for review, and it will be publicly available upon acceptance.

Declarations

Conflict of interest

The authors declare no conflict of interests.

References

[1] Taketsugu, H., Oba, T., Maeda, T., Nobuhara, S., Ukita, N.: Physical plausibility-aware trajectory prediction via locomotion embodiment. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp. 12324–12334 (2025). https://doi.org/10.1109/CVPR52734.2025.01150

[2] Kress, V., Jeske, F., Zernetsch, S., Doll, K., Sick, B.: Pose and semantic map based probabilistic forecast of vulnerable road users' trajectories. IEEE Trans. Intell. Veh. **8**(3), 2592–2603 (2023) https://doi.org/10.1109/TIV.2022.3149624

[3] Kothari, P., Kreiss, S., Alahi, A.: Human trajectory forecasting in crowds: A deep learning perspective. IEEE Trans. Intell. Transp. Syst. **23**(7), 7386–7400 (2022) https://doi.org/10.1109/TITS.2021.3069362

[4] Gao, Y., Saadatnejad, S., Alahi, A.: Social-pose: Enhancing trajectory prediction with human body pose. IEEE Trans. Intell. Transp. Syst. **26**(11), 21309–21319 (2025) https://doi.org/10.1109/TITS.2025.3594889

[5] Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social LSTM: human trajectory prediction in crowded spaces. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp. 961–971 (2016). https://doi.org/10.1109/CVPR.2016.110

[6] Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social GAN: socially acceptable trajectories with generative adversarial networks. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp. 2255–2264 (2018). https://doi.org/10.1109/CVPR.2018.00240

[7] Saadatnejad, S., Gao, Y., Messaoud, K., Alahi, A.: Social-transmotion: Promptable human trajectory prediction. In: Proc. Int. Conf. Learn. Represent. (2024)

[8] Xu, C., Tan, R.T., Tan, Y., Chen, S., Wang, Y.G., Wang, X., Wang, Y.: EqMotion: Equivariant multi-agent motion prediction with invariant interaction reasoning. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp. 1410–1420 (2023). https://doi.org/10.1109/CVPR52729.2023.00142

[9] Helbing, D., Molnár, P.: Social force model for pedestrian dynamics. Phys. Rev. E **51**(5), 4282–4286 (1995) https://doi.org/10.1103/PhysRevE.51.4282

[10] Yue, J., Manocha, D., Wang, H.: Human trajectory prediction via neural social physics. In: Proc. Eur. Conf. Comput. Vis., pp. 376–394 (2022). https://doi.org/10.1007/978-3-031-19830-4_22

[11] Salzmann, T., Ivanovic, B., Chakravarty, P., Pavone, M.: Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In: Proc. Eur. Conf. Comput.

Vis., pp. 683–700 (2020). https://doi.org/10.1007/978-3-030-58523-5_40

[12] Cui, H., Nguyen, T., Chou, F.-C., Lin, T.-H., Schneider, J., Bradley, D., Djuric, N.: Deep kinematic models for kinematically feasible vehicle trajectory predictions. In: Proc. IEEE Int. Conf. Robot. Autom., pp. 10563–10569 (2020). https://doi.org/10.1109/ICRA40945.2020.9197560

[13] Sun, J., Li, Y., Chai, L., Fang, H.-S., Li, Y.-L., Lu, C.: Human trajectory prediction with momentary observation. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp. 6457–6466 (2022). https://doi.org/10.1109/CVPR52688.2022.00636

[14] Xu, Y., Fu, Y.: Adapting to length shift: Flexilength network for trajectory prediction. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp. 15226–15237 (2024). https://doi.org/10.1109/CVPR52733.2024.01442

[15] Adeli, V., Ehsanpour, M., Reid, I., Niebles, J.C., Savarese, S., Adeli, E., Rezatofighi, H.: TRiPOD: Human trajectory and pose dynamics forecasting in the wild. In: Proc. IEEE/CVF Int. Conf. Comput. Vis., pp. 13370–13380 (2021). https://doi.org/10.1109/ICCV48922.2021.01314

[16] Salzmann, T., Chiang, H.-T.L., Ryll, M., Sadigh, D., Parada, C., Bewley, A.: Robots that can see: Leveraging human pose for trajectory prediction. IEEE Robot. Autom. Lett. **8**(11), 7090–7097 (2023) https://doi.org/10.1109/LRA.2023.3312035

[17] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997) https://doi.org/10.1162/neco.1997.9.8.1735

[18] Zhang, X., Angeloudis, P., Demiris, Y.: Dual-branch spatio-temporal graph neural networks for pedestrian trajectory prediction. Pattern Recognit. **142**, 109633 (2023) https://doi.org/10.1016/J.PATCOG.2023.109633

[19] Bouzayane, S., Kahouadji, M., Magnier, B.: Fort-raj: a fisheye-optimized deep learning model for real-time trajectory prediction. Pattern Analysis and Applications **28**(2), 114 (2025) https://doi.org/10.1007/s10044-025-01485-y

[20] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Proc. Adv. Neural Inf. Process. Syst., pp. 6000–6010 (2017)

[21] Nishimura, M., Nobuhara, S., Nishino, K.: Viewbirdiformer: Learning to recover ground-plane crowd trajectories and ego-motion from a single ego-centric view. IEEE Robot. Autom. Lett. **8**(1), 368–375 (2023) https://doi.org/10.1109/LRA.2022.3221335

[22] Giuliari, F., Hasan, I., Cristani, M., Galasso, F.: Transformer networks for trajectory forecasting. In: Proc. Int. Conf. Pattern Recognit., pp. 10335–10342 (2021). https://doi.org/10.1109/ICPR48806.2021.9412190

[23] Yuan, Y., Weng, X., Ou, Y., Kitani, K.: AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting. In: Proc. IEEE/CVF Int. Conf. Comput. Vis., pp. 9793–9803 (2021). https://doi.org/10.1109/ICCV48922.2021.00967

[24] Yang, T., Wang, G., Lai, J., Wang, Y.: A new multi-feature fusion trajectory prediction method. Pattern Analysis and Applications **28**(4), 168 (2025) https://doi.org/10.1007/s10044-025-01541-7

[25] Peng, X., Mao, S., Wu, Z.: Trajectory-aware body interaction transformer for multi-person pose forecasting. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp. 17121–17130 (2023). https://doi.org/10.1109/CVPR52729.2023.01642

[26] Jeong, J., Lee, S., Park, D., Lee, G., Yoon, K.-J.: Multi-modal knowledge distillation-based human trajectory forecasting. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp. 24222–24233 (2025). https://doi.org/10.1109/CVPR52734.2025.02256

[27] Jeong, J., Park, D., Yoon, K.-J.: Multi-agent long-term 3d human pose forecasting via interaction-aware trajectory conditioning. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp. 16975–16984 (2024). https://doi.org/10.1109/CVPR52733.2024.00160

[28] Gao, Y., Luan, P., Alahi, A.: Multi-transmotion: Pre-trained model for human motion prediction. In: Proc. Conf. Robot Learn. (2024)

[29] Peng, X.B., Abbeel, P., Levine, S., Panne, M.: Deepmimic: example-guided deep reinforcement learning of physics-based character skills. ACM Trans. Graph. **37**(4), 1–14 (2018) https://doi.org/10.1145/3197517.3201311

[30] Peng, X.B., Ma, Z., Abbeel, P., Levine, S., Kanazawa, A.: Amp: adversarial motion priors for stylized physics-based character control. ACM Trans. Graph. **40**(4), 1–20 (2021) https://doi.org/10.1145/3450626.3459670

[31] Rempe, D., Luo, Z., Peng, X.B., Yuan, Y., Kitani, K., Kreis, K., Fidler, S., Litany, O.: Trace and pace: Controllable pedestrian animation via guided trajectory diffusion. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp. 13756–13766 (2023). https://doi.org/10.1109/CVPR52729.2023.01322

[32] Rasouli, A., Kotseruba, I., Kunic, T., Tsotsos, J.: PIE: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction. In: Proc. IEEE/CVF Int. Conf. Comput. Vis., pp. 6261–6270 (2019). https://doi.org/10.1109/ICCV.2019.00636

[33] Rasouli, A., Kotseruba, I., Tsotsos, J.K.: Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior. In: Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops, pp. 206–213 (2017). https://doi.org/10.1109/ICCVW.2017.33

[34] Zhou, Y., Tan, G., Zhong, R., Li, Y., Gou, C.: PIT: Progressive interaction transformer for pedestrian crossing intention prediction. IEEE Trans. Intell. Transport. Sys. **24**(12), 14213–14225 (2023) https://doi.org/10.1109/TITS.2023.3309309

[35] Mahjourian, R., Kim, J., Chai, Y., Tan, M., Sapp, B., Anguelov, D.: Occupancy flow fields for motion forecasting in autonomous driving. IEEE Robot. Autom. Lett. **7**(2), 5639–5646 (2022) https://doi.org/10.1109/LRA.2022.3151613

[36] Xu, H., Peng, P., Zhang, X., Tan, G., Li, Y., Wang, S., Li, L.: Exploiting continuous motion clues for vision-based occupancy prediction. In: Proc. AAAI Conf. Artif. Intell., pp. 8860–8868 (2025). https://doi.org/10.1609/aaai.v39i8.32958

[37] Wang, X., Zhu, Z., Huang, G., Chen, X., Zhu, J., Lu, J.: Drivedreamer: Towards real-world-drive world models for autonomous driving, pp. 55–72 (2024). https://doi.org/10.1007/978-3-031-73195-2_4

[38] Wang, Y., He, J., Fan, L., Li, H., Chen, Y., Zhang, Z.: Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp. 14749–14759. IEEE, ??? (2024). https://doi.org/10.1109/CVPR52733.2024.01397 . https://doi.org/10.1109/CVPR52733.2024.01397

[39] Li, Y., Wang, S., Guo, M., Huang, J., Ding, T., Hu, M., Wang, K., Shen, S., Tan, G.: ReCamDriving: LiDAR-Free Camera-Controlled Novel Trajectory Video Generation (2025). https://doi.org/10.48550/arXiv.2512.03621

[40] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)

[41] Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A., State, G.: Isaac gym: High performance gpu-based physics simulation for robot learning. In: Proc. Adv. Neural Inf. Process. Syst.

Datasets and Benchmarks Track (2021)

[42] Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.: Amass: Archive of motion capture as surface shapes. In: Proc. IEEE/CVF Int. Conf. Comput. Vis., pp. 5441–5450 (2019). https://doi.org/10.1109/ICCV.2019.00554

[43] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for NLP. In: Proc. Int. Conf. Mach. Learn., vol. 97, pp. 2790–2799 (2019)

[44] Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. In: Proc. Int. Conf. Learn. Represent. (2022)

[45] Fujii, R., Saito, H., Hachiuma, R.: Towards predicting any human trajectory in context. In: Proc. Adv. Neural Inf. Process. Syst. (2025)

[46] Martín-Martín, R., Patel, M., Rezatofighi, H., Shenoi, A., Gwak, J., Frankel, E., Sadeghian, A., Savarese, S.: JRDB: A dataset and benchmark of egocentric robot visual perception of humans in built environments. IEEE Trans. Pattern Anal. Mach. Intell. **45**(6), 6748–6765 (2023) https://doi.org/10.1109/TPAMI.2021.3070543

[47] Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: a skinned multi-person linear model. ACM Trans. Graph. **34**(6), 1–16 (2015) https://doi.org/10.1145/2816795.2818013

[48] Fabbri, M., Lanzi, F., Calderara, S., Palazzi, A., Vezzani, R., Cucchiara, R.: Learning to detect and track visible and occluded body joints in a virtual world. In: Proc. Eur. Conf. Comput. Vis., pp. 450–466 (2018). https://doi.org/10.1007/978-3-030-01225-0_27

[49] Xu, Z., Zhou, R., Yin, Y., Gao, H., Tomizuka, M., Li, J.: MATRIX: Multi-agent trajectory generation with diverse contexts. In: Proc. IEEE Int. Conf. Robot. Autom., pp. 12650–12657 (2024). https://doi.org/10.1109/ICRA57147.2024.10610944

[50] Girgis, R., Golemo, F., Codevilla, F., Weiss, M., D'Souza, J.A., Kahou, S.E., Heide, F., Pal, C.: Latent variable sequential set transformers for joint multi-agent motion prediction. In: Proc. Int. Conf. Learn. Represent. (2022)

[51] Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp. 7482–7491 (2018). https://doi.org/10.1109/CVPR.2018.00781

[52] Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., Finn, C.: Gradient surgery for multi-task learning. In: Proc. Adv. Neural Inf. Process. Syst., pp. 5824–5836 (2020). https://doi.org/10.5555/3495724.3496213

[53] Liu, S., Chang, P., Huang, Z., Chakraborty, N., Hong, K., Liang, W., McPherson, D.L., Geng, J., Driggs-Campbell, K.R.: Intention aware robot crowd navigation with attention-based interaction graph. In: Proc. IEEE Int. Conf. Robot. Autom., pp. 12015–12021 (2023). https://doi.org/10.1109/ICRA48891.2023.10160660