# Enhancing Trajectory Prediction through Self-Supervised Waypoint Distortion Prediction

**Pranav Singh Chib** [*] [1] **Pravendra Singh** [*] [1]

## Abstract

Trajectory prediction is an important task that involves modeling the indeterminate nature of agents to forecast future trajectories given the observed trajectory sequences. The task of predicting trajectories poses significant challenges, as agents not only move individually through time but also interact spatially. The learning of complex spatio-temporal representations stands as a fundamental challenge in trajectory prediction. To this end, we propose a novel approach called SSWDP (Self-Supervised Waypoint Distortion Prediction). We propose a simple yet highly effective self-supervised task of predicting distortion present in the observed trajectories to improve the representation learning of the model. Our approach can complement existing trajectory prediction methods. The experimental results highlight a significant improvement with relative percentage differences of 22.7%/38.9%, 33.8%/36.4%, and 16.60%/23.20% in ADE/FDE for the NBA, TrajNet++, and ETH-UCY datasets, respectively, compared to the baseline methods. Our approach also demonstrates a significant improvement over baseline methods with relative percentage differences of 76.8%/82.5% and 61.0%/36.1% in ADE/FDE for TrajNet++ and NBA datasets in distorted environments, respectively.

## 1. Introduction

Trajectory prediction involves estimating an agent's future movement by analyzing its historical past trajectories. This process holds significant importance in various applications, including autonomous driving, robotics, surveillance systems, drones, and other autonomous systems. Several research studies have focused on utilizing deep generative models (Gu et al., 2022; Mao et al., 2023; Xu et al., 2022a) to address trajectory prediction tasks. For example, some approaches employ generative adversarial networks (GANs) (Gupta et al., 2018; Hu et al., 2020; Sadeghian et al., 2019) to diversify the distribution across all potential future trajectories. Other approaches (Lee et al., 2022; Mangalam et al., 2020; Xu et al., 2022a;c) utilize conditional variational autoencoders (CVAE) to capture the multi-modal distribution of future trajectories. Transformers (Girgis et al., 2022; Giuliari et al., 2021; Gu et al., 2023; Tsao et al., 2022) have also been employed for trajectory prediction. Several works (Bae & Jeon, 2023; Lv et al., 2023; Sekhon & Fleming, 2021; Xu et al., 2022a; 2023c) use graph-based models to model the social interactions between agents in trajectory prediction tasks.

The future trajectories of agents, such as pedestrians, often exhibit uncertainty due to the ability of these agents to adapt their movement in response to changing environments and physical constraints. The task of predicting trajectories poses significant challenges, as agents move not only individually through time but also interact spatially. Learning complex spatio-temporal representations stands as a fundamental challenge in trajectory prediction. Consequently, an effective trajectory forecasting model must be capable of learning the underlying representation more effectively. Some works attempt data augmentation techniques to help the model in learning better representations. Data augmentations, such as trajectory flipping (Ye et al., 2023), trajectory masking (Chen et al., 2023b; Cheng et al., 2023), and noise augmentation (adding noise to trajectory) (Bae et al., 2023; Saadatnejad et al., 2022; Ye et al., 2023), have been explored. However, these augmentations have not yielded significant improvements, as evident in the S-ATTack study (Saadatnejad et al., 2022) where noise augmentation did not lead to a significant improvement. We also attempted flipping, masking, and noise augmentations using the GroupNet baseline (Xu et al., 2022a) over the NBA dataset and did not obtain a significant improvement compared to the baseline model (see Section 4.6.1). Therefore, in this work, we focus on utilizing self-supervised learning. Self-supervision (Wei et al., 2019) has garnered significant attention, and its

[*]Equal contribution [1]Department of Computer Science and Engineering, Indian Institute of Technology, Roorkee, India. Correspondence to: Pranav Singh Chib <pranavs_chib@cs.iitr.ac.in>, Pravendra Singh <pravendra.singh@cs.iitr.ac.in>.

*Proceedings of the $41^{st}$ International Conference on Machine Learning*, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

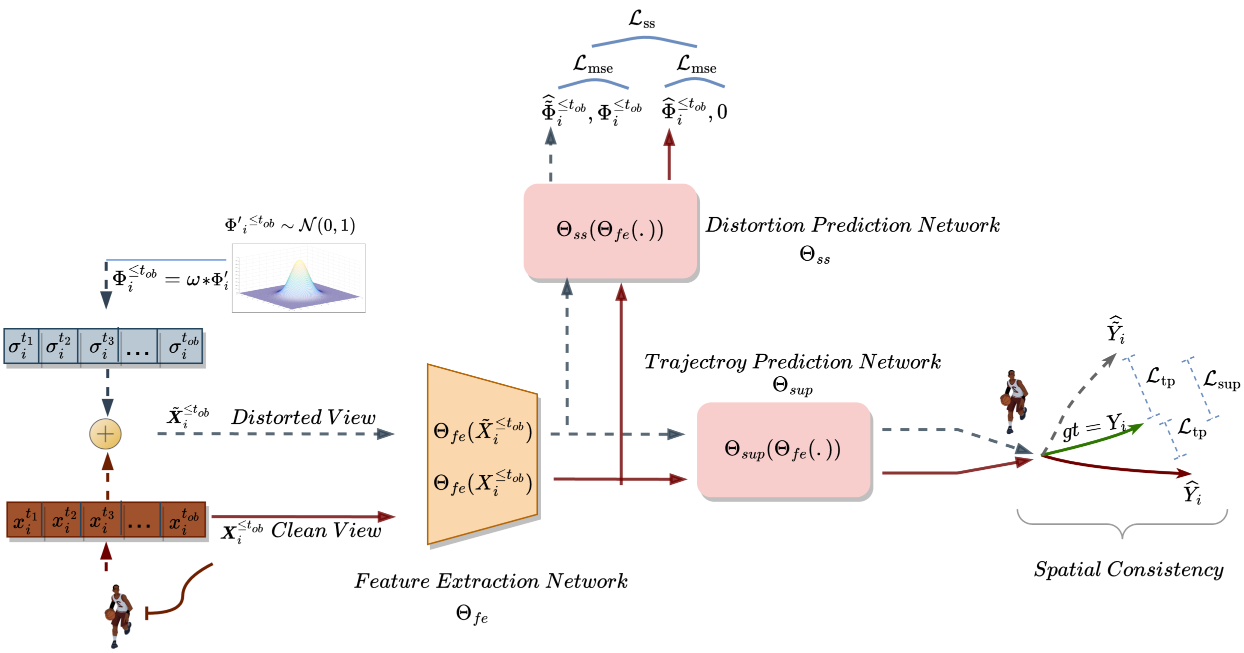
$\mathcal{L}_{\text{ss}}$
$\mathcal{L}_{\text{mse}}$
$\mathcal{L}_{\text{mse}}$
$\widehat{\tilde{\Phi}}_i^{\le t_{ob}}, \Phi_i^{\le t_{ob}}$
$\widehat{\Phi}_i^{\le t_{ob}}, 0$
$\Theta_{ss}(\Theta_{fe}(.))$
Distortion Prediction Network
$\Theta_{ss}$
${\Phi'}_i^{\le t_{ob}} \sim \mathcal{N}(0,1)$
$\Phi_i^{\le t_{ob}} = \omega * \Phi'_i$
$\sigma_i^{t_1}$ $\sigma_i^{t_2}$ $\sigma_i^{t_3}$ ... $\sigma_i^{t_{ob}}$
$\tilde{X}_i^{\le t_{ob}}$ Distorted View
$\Theta_{fe}(\tilde{X}_i^{\le t_{ob}})$
$\Theta_{fe}(X_i^{\le t_{ob}})$
Trajectroy Prediction Network
$\Theta_{sup}$
$\Theta_{sup}(\Theta_{fe}(.))$
$x_i^{t_1}$ $x_i^{t_2}$ $x_i^{t_3}$ ... $x_i^{t_{ob}}$
$X_i^{\le t_{ob}}$ Clean View
Feature Extraction Network
$\Theta_{fe}$
$\widehat{\tilde{Y}}_i$
$\mathcal{L}_{\text{tp}}$
$\mathcal{L}_{\text{sup}}$
$gt = Y_i$
$\mathcal{L}_{\text{tp}}$
$\widehat{Y}_i$
Spatial Consistency

*Figure 1.* Illustration of SSWDP, where we first create two distinct views of past observed trajectories: the clean view $\boldsymbol{X}_i^{\le t_{ob}}$ and the distortion view $\tilde{\boldsymbol{X}}_i^{\le t_{ob}}$ across the spatial domain of waypoints. We then enforce the trajectory prediction model $\Theta_{sup}$ to maintain spatial consistency between predictions from these two views and predict the future trajectories $\widehat{Y}_i$ and $\widehat{\tilde{Y}}_i$ (see Section 3.2.2). In the distortion prediction module, we predict the distortion present in the two views of past observed trajectories as an auxiliary self-supervised pretext task (see Section 3.2.3). Brown and gray lines represent the flow for clean and distorted views, respectively, while the green lines represent the ground truth ($gt$).

objective is to assist models in acquiring more generalized representations through pretext tasks.

To enhance representation learning, we propose a novel approach called SSWDP (Self-Supervised Waypoint Distortion Prediction). We propose a simple yet highly effective self-supervised pretext task: predicting distortion present in the observed trajectories. To the best of our knowledge, this paper is the first work to explore distortion prediction as a self-supervised pretext task to improve the representation learning of the model in trajectory prediction. Our approach consists of two modules (spatial consistency module and distortion prediction module) as shown in Figure 1. In the spatial consistency module, we create two distinct views of past observed trajectories: the clean view and the distorted view. The clean view represents the original past observed trajectory, while the distorted view represents past observed trajectories that have been spatially relocated. Since the distorted view is created by relocating waypoints from the clean view, both views are similar (though not identical). We enforce the trajectory prediction model to maintain spatial consistency between predictions derived from these two views.

In the distortion prediction module, we predict the spatial distortion present in the past observed trajectories (coordinates) as an auxiliary self-supervised pretext task. This simple yet highly effective task assists the trajectory prediction model in better learning the underlying representation for trajectory prediction, thereby enhancing future predictions (see Table 3 and Figure 2). It is important to note that the novelty of our work lies in the proposed pretext task, not the augmentation. We demonstrated in Section 4.6.1 that using only augmentation approaches results in suboptimal performance. We also conduct ablation experiments in Section 4.6.2 to empirically demonstrate that both modules (i.e., spatial consistency module and distortion prediction module) are crucial for our approach.

Our approach can be easily integrated with existing trajectory prediction methods. We have integrated our approach into four existing methods: the generative-based GroupNet (Xu et al., 2022a), the goal-oriented Graph-TERN (Bae & Jeon, 2023), Graph-based SSAGCN (Lv et al., 2023) and the transformer-based AutoBot (Girgis et al., 2022). Our extensive experiments demonstrate the ability of our approach to accurately forecast future trajectories, leading to substantial performance improvements across the NBA SportVU (Zhan et al., 2018), synthetic partition Trajnet++ (Kothari et al., 2021), and ETH-UCY (Pellegrini et al., 2009) datasets.

Additionally, we conduct ablation experiments (see Section 4.6.4) to demonstrate that incorporating SSWDP into the model learning process significantly improves performance in a distorted environment compared to the baseline method.

## 2. Related Work

### 2.1. Trajectory prediction

The trajectory forecasting model seeks to predict future trajectories considering the observed trajectories. When forecasting an agent's future trajectory, a wide range of future trajectories can be possible. The stochastic prediction model has been used in several works (Gupta et al., 2018; Lee et al., 2022; Mangalam et al., 2020; Shi et al., 2021; Xu et al., 2022c). These models include a range of methods, such as conditional variational autoencoders (CVAEs) (Lee et al., 2022; Mangalam et al., 2020; Xu et al., 2022a;c), generative adversarial networks (GANs) (Gupta et al., 2018; Hu et al., 2020; Sadeghian et al., 2019), and diffusion models (Gu et al., 2022; Mao et al., 2023). Some work, including RMB (Shi et al., 2023a), addresses the issue of superfluous interactions by proposing the Interpretable Multimodality Predictor (IMP), which models the distribution of mean locations as a Gaussian Mixture Model (GMM) and encourages multimodality by sampling multiple mean locations of predictions. Transformers (Girgis et al., 2022; Yu et al., 2020; Yuan et al., 2021; Zhou et al., 2023) are widely employed to capture temporal and social dimensions through the attention mechanism. Graph-based (Lv et al., 2023; Sekhon & Fleming, 2021) methods are specifically utilized to explicitly model social interactions among agents in the scene through relational reasoning. These approaches adeptly capture interactions and their associated strengths in both groupwise and pairwise interactions to predict plausible future trajectories. DynGroupNet (Xu et al., 2023b) and TDGCN (Wang et al., 2023a) focus on capturing temporal groupwise interactions, considering interaction strength and interaction category. Additionally, several other efforts have been made in trajectory prediction, such as endpoint-conditioned trajectory prediction (Bae & Jeon, 2023), long-tail trajectory prediction (Wang et al., 2023b), and others. MERA (Sun et al., 2023) utilizes different types of modalities in motion predictions, processing different feature clusters to represent modalities such as scene semantics and agent motion state.

### 2.2. Self-supervised Learning

Self-supervised learning is a paradigm that has gained popularity across various domains of deep learning, including computer vision. Through different pretext tasks, additional supervision is generated from unlabeled data, which is then used to train a model in a self-supervised manner. Several self-supervised approaches (Caron et al., 2018; Gidaris et al., 2018; Pathak et al., 2017; Wei et al., 2019) have been developed to acquire better representation learning. For example, in the context of acquiring image features (Gidaris et al., 2018), self-supervised tasks train deep networks to recognize the 2D rotation angles of images. In another approach (Wei et al., 2019), a pretext task is proposed to learn spatial relationships by dividing an image into a grid of patches, rearranging their spatial positions, and training the network to restore their accurate spatial arrangement. Additional self-supervised learning techniques include image clustering (Caron et al., 2018), segmentation prediction (Pathak et al., 2017), and others.

Recently, in trajectory prediction (Bhattacharyya et al., 2023; Halawa et al., 2022; Wang et al., 2023b), a few works have explored self-supervised learning. Some employ contrastive learning (Halawa et al., 2022; Wang et al., 2023b) to enhance the representation ability of the network, while others, like SSL lanes (Bhattacharyya et al., 2023), utilize map/agent-level data to formulate various pretext tasks. Unlike the above-mentioned methods, we propose a novel pretext task that predicts the distortion present in the clean and distorted views of past observed trajectories as an auxiliary self-supervised pretext task to enhance the trajectory prediction task.

### 2.3. Learning with Regularization

Several techniques have been explored in recent studies to regularize trajectory prediction (Chen et al., 2023b; Cheng et al., 2023; Saadatnejad et al., 2022; Ye et al., 2023). Some methodologies, like the one proposed by Ye et al. (Ye et al., 2023), utilize a variety of transformations applied to the same input data to generate perturbation-invariant representations. Wu et al. (Wu et al., 2023) employed masked trajectory predictions and reconstruction to extract additional pieces of information from trajectories. TENET (Wang et al., 2022b) propagates learning embeddings through a temporal flow network to reconstruct the input, serving as a means to enhance the acquired embeddings. Researchers (Girgis et al., 2022; Sekhon & Fleming, 2021; Zhu et al., 2020) also implement a mechanism wherein the predicted future trajectory is reversed temporally and fed back into the prediction model. This approach aims to predict the historical trajectory, and the loss is calculated with the inclusion of an extra cycle loss term.

## 3. Methodology

### 3.1. Problem Formulation

The goal of trajectory predictions is to forecast the future trajectories of agents in a dynamic environment based on their past trajectories. A trajectory is represented by a temporal series of spatial points, termed as waypoints. The past observed trajectory, spanning from $t_1$ to $t_{ob}$, can be denoted as $X_i^{\leq t_{ob}} = \{\boldsymbol{x}_i^{t_1}, \boldsymbol{x}_i^{t_2}, ..., \boldsymbol{x}_i^{t_{ob}}\}$, where $\boldsymbol{x}_i^{t_{ob}} \in \mathbb{R}^2$ corresponds to the 2D coordinates of agent $i$ at time step $t_{ob}$. Similarly, the predicted future trajectory for agent $i$ over the duration $t_{ob+1}$ to $t_{fu}$ can be described as $\widehat{Y}_i^{t_{ob+1} \leq t \leq t_{fu}}$. Corresponding ground truth for the future trajectory of agent $i$ can be described as $Y_i^{t_{ob+1} \leq t \leq t_{fu}}$ over the duration $t_{ob+1}$ to $t_{fu}$. It is worth noting that the problem formulation is given for the $i^{th}$ agent for the sake of simplicity, but it can be generalized to all agents present in the scene.

## 3.2. Self-Supervised Waypoint Distortion Prediction

### 3.2.1. Clean and Distorted Views

In our approach, we first generate two different views of past observed trajectories: one characterized as the clean view and the other as a distorted view. The clean view corresponds to the original past trajectory, while the distorted view corresponds to the past trajectory that has been spatially relocated with some additive noise.

Given the observed past trajectory $X_i^{\leq t_{ob}}$ of agent $i$, the clean view and distorted view are denoted by $X_i^{\leq t_{ob}}$ and $\tilde{X}_i^{\leq t_{ob}}$ respectively. We sample the Gaussian noise ($\mathcal{N}(0,1)$) and add it to $X_i^{\leq t_{ob}}$ to create the distorted view. Specifically, $\Phi'^{\leq t_{ob}}_i \sim \mathcal{N}(0,1)$ is noise sampled from the standard normal distribution. We control this noise by a parameter ($\omega$) called the noise factor to get the final additive noise ($\Phi_i^{\leq t_{ob}}$) as shown in Eq. 1. We have also provided an ablation in Section 4.6.3 on choosing the appropriate $\omega$ value. The noise factor controls the spatial relocation of waypoints in the distorted view.

$$\Phi_i^{\leq t_{ob}} = \omega * \Phi'^{\leq t_{ob}}_i \tag{1}$$

$$\tilde{\boldsymbol{X}}_i^{\leq t_{ob}} = \boldsymbol{X}_i^{\leq t_{ob}} + \boldsymbol{\Phi}_i^{\leq t_{ob}} \tag{2}$$

$$\tilde{\boldsymbol{X}}_i^{\leq t_{ob}} = \{\boldsymbol{x}_i^{t_1}, ..., \boldsymbol{x}_i^{t_{ob}}\} + \{\boldsymbol{\sigma}_i^{t_1}, ..., \boldsymbol{\sigma}_i^{t_{ob}}\} \tag{3}$$

We add additive noise $\boldsymbol{\Phi}_i^{\leq t_{ob}} = \{\boldsymbol{\sigma}_i^{t_1}, ..., \boldsymbol{\sigma}_i^{t_{ob}}\}$ to the past observed trajectory $\boldsymbol{X}_i^{\leq t_{ob}} = \{\boldsymbol{x}_i^{t_1}, ..., \boldsymbol{x}_i^{t_{ob}}\}$ of agent $i$ to obtain the distorted view ($\tilde{\boldsymbol{X}}_i^{\leq t_{ob}}$). Here, $\boldsymbol{\sigma}_i^{t_{ob}} \in \mathbb{R}^2$ represents the 2D Gaussian noise vector for agent $i$ at time step $t_{ob}$.

### 3.2.2. Spatial Consistency Module

After creating clean and distorted views for agent $i$, we feed them as input to the feature extraction network ($\Theta_{fe}$). The feature extraction network generates features corresponding to both the clean view and the distorted view. The features from the clean view are then input into the trajectory prediction network ($\Theta_{sup}$) to predict the future trajectory ($\widehat{Y}_i$). Similarly, the features from the distorted view are also passed through the trajectory prediction network ($\Theta_{sup}$) to obtain the future trajectory corresponding to the distorted view, as indicated in the Eqs. below.

$$\widehat{Y}_i^{t_{ob+1} \leq t \leq t_{fu}} = \Theta_{sup}(\Theta_{fe}(X_i^{\leq t_{ob}})) \tag{4}$$

$$\widehat{\tilde{Y}}_i^{t_{ob+1} \leq t \leq t_{fu}} = \Theta_{sup}(\Theta_{fe}(\tilde{X}_i^{\leq t_{ob}}) \tag{5}$$

Here, $\widehat{Y}_i$ and $\widehat{\tilde{Y}}_i$ denote the future trajectory predictions from the clean and distorted views of the past observed trajectory, respectively. Next, we use the trajectory prediction loss ($\mathcal{L}_{\text{tp}}$) to minimize the gap between the predicted trajectory and the ground truth trajectory. The supervised loss ($L_{\text{sup}}$) is calculated using Eq. 6. It is evident from Eq. 6 that we are minimizing the gap between $\widehat{Y}_i$ and $Y_i$. Simultaneously, we are also minimizing the gap between $\widehat{\tilde{Y}}_i$ and $Y_i$, thus implicitly minimizing the gap between $\widehat{Y}_i$ and $\widehat{\tilde{Y}}_i$. Therefore, $\mathcal{L}_{\text{sup}}$ maintains spatial consistency between the future predictions from clean observed trajectories and the distorted trajectories. This consistency ensures that both views are consistent with each other.

$$\begin{aligned}\mathcal{L}_{\text{sup}} = \frac{1}{N}\sum_{i=1}^{N}\Big(&\mathcal{L}_{\text{tp}}(\widehat{Y}_i^{t_{ob+1} \leq t \leq t_{fu}}, Y_i^{t_{ob+1} \leq t \leq t_{fu}}) + \\ &\mathcal{L}_{\text{tp}}(\widehat{\tilde{Y}}^{t_{ob+1} \leq t \leq t_{fu}}, Y_i^{t_{ob+1} \leq t \leq t_{fu}})\Big)\end{aligned} \tag{6}$$

Where $N$ is the number of agents, $Y_i$ is the ground truth future trajectory for agent $i$.

### 3.2.3. Distortion Prediction Module

The self-supervised distortion prediction task involves predicting the distortion present in both the clean view (observed past trajectory $X_i^{\leq t_{ob}}$) and the distorted view ($\tilde{X}_i^{\leq t_{ob}}$). Specifically, the goal is to estimate the spatial relocation values associated with the given observed waypoints.

$$\begin{aligned}\widehat{\tilde{\Phi}}_i^{\leq t_{ob}} &= \Theta_{ss}(\Theta_{fe}(\tilde{X}_i^{\leq t_{ob}})) \\ \widehat{\Phi}_i^{\leq t_{ob}} &= \Theta_{ss}(\Theta_{fe}(X_i^{\leq t_{ob}}))\end{aligned} \tag{7}$$

Where $\Theta_{ss}$ represents the parameters of the distorted prediction network. $\widehat{\tilde{\Phi}}_i^{\leq t_{ob}}$ is the predicted spatial relocation values in the distorted view of agent $i$, $\widehat{\Phi}_i^{\leq t_{ob}}$ is the predicted spatial relocation values for the clean view of agent $i$. $\Phi_i^{\leq t_{ob}}$ is the ground truth relocation values (see Eq. 1). Please note that the features extracted by $\Theta_{fe}$ are utilized as input to $\Theta_{ss}$ (distortion predicting network) for predicting the relocation values in the observed past trajectory (clean and distorted views as shown in Eq. 7).

The self supervised distortion prediction loss ($\mathcal{L}_{\text{ss}}$) is given

*Table 1.* The Average Displacement Error (ADE) and Final Displacement Error (FDE) for prediction on the NBA dataset using the *SSWDP* approach. (B) denotes the baseline GroupNet model. RD(%) indicates the relative percentage difference compared to the baseline. The top performance is highlighted in **bold**.

| Method | Time | | | | |
|---|---|---|---|---|---|
| | Venue | 1.0s | 2.0s | 3.0s | 4.0s |
| SSTGCNN | CVPR 2020 | 0.36/0.50 | 0.75/0.99 | 1.15/1.79 | 1.59/2.37 |
| STAR | ECCV 2020 | 0.43/0.65 | 0.77/1.28 | 1.00/1.55 | 1.26/2.04 |
| PECNet | ECCV 2020 | 0.51/0.76 | 0.96/1.69 | 1.41/2.52 | 1.83/3.41 |
| NMMP | CVPR 2020 | 0.38/0.54 | 0.70/1.11 | 1.01/1.61 | 1.33/2.05 |
| MemoNet | CVPR 2022 | 0.38/0.56 | 0.71/1.14 | 1.00/1.57 | 1.25/1.47 |
| NPSN | CVPR 2022 | 0.35/0.58 | 0.68/1.23 | 1.01/1.76 | 1.31/1.79 |
| GroupNet (B) | CVPR 2022 | 0.34/0.48 | 0.62/0.95 | 0.87/1.31 | 1.13/1.69 |
| MERA | TPAMI 2023 | - | - | - | 1.17/2.21 |
| DISTL | NeurIPS 2023 | 0.30/0.40 | 0.58/0.88 | 0.87/1.31 | 1.13/1.60 |
| DCG | arXiv 2024 | - | - | - | 1.16/1.64 |
| Our (B)+SSWDP | - | **0.23/0.31** | **0.45/0.63** | **0.67/0.92** | **0.90/1.14** |
| RD(%) ADE/FDE | - | 38.6/43.0 | 31.8/40.5 | 26.0/35.0 | 22.7/38.9 |

below:

$$\mathcal{L}_{\text{ss}} = \frac{1}{N}\sum_{i=1}^{N}\Big(\mathcal{L}_{\text{mse}}(\widehat{\Phi}_i^{\leq t_{ob}}, 0) + \mathcal{L}_{\text{mse}}(\widehat{\widetilde{\Phi}}_i^{\leq t_{ob}}, \Phi_i^{\leq t_{ob}})\Big) \tag{8}$$

Here, mse refers to mean square error. 0 signifies that the spatial relocation values are zero in the clean view, indicating that no distortion is present in the original past observed trajectory ($X_i^{\leq t_{ob}}$) of agent $i$.

It is worth noting from Eq. 8 that we are also minimizing $\mathcal{L}_{\text{mse}}(\widehat{\Phi}_i^{\leq t_{ob}}, 0)$ along with $\mathcal{L}_{\text{mse}}(\widehat{\widetilde{\Phi}}_i^{\leq t_{ob}}, \Phi_i^{\leq t_{ob}})$ so that the model can better differentiate between non-distorted and distorted trajectories.

### 3.3. Learning and Evaluation

The total loss is defined using Eq. 9.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{sup}} + \lambda\mathcal{L}_{\text{ss}} \tag{9}$$

$$\mathbf{\Theta}^{\star}_{\mathbf{fe}}, \mathbf{\Theta}^{\star}_{\mathbf{sup}}, \mathbf{\Theta}^{\star}_{\mathbf{ss}} = \underset{\mathbf{\Theta}_{\mathbf{fe}}, \mathbf{\Theta}_{sup}, \mathbf{\Theta}_{ss}}{\text{arg min}} \quad \mathcal{L}_{\text{total}} \tag{10}$$

Here, $\mathcal{L}_{\text{total}}(\cdot)$ denotes the total loss for training the SSWDP. $\lambda$ signifies the contribution of the self-supervised loss in the total loss for training the model using our approach. Using Eq. 10, we get the optimal parameters that would be utilized at inference time as given in Eq. 11. At evaluation time, we can predict the future trajectory using Eq. 11 for any given observed trajectory.

$$\widehat{Y}_i^{t_{ob+1} \leq t \leq t_{fu}} = \Theta^{\star}_{\text{sup}}(\Theta^{\star}_{\text{fe}}(X_i^{\leq t_{ob}})) \tag{11}$$

## 4. Experiments

In this section, we present the quantitative and qualitative results of our approach. Additionally, we have conducted several ablation studies to validate our approach. Experimental details, including implementation details, baseline models, and architecture details, have been provided in Appendix A.

### 4.1. Datasets

We evaluate the performance of SSWDP on three trajectory datasets: NBA (Zhan et al., 2018), a synthetic partition of TrajNet++ (Kothari et al., 2021), and ETH-UCY (Lerner et al., 2007; Pellegrini et al., 2009). The NBA Sports VU Dataset includes player trajectory data from all ten players in live NBA games, where teammates heavily influence player motions. In this assessment, we predict the following ten timestamps (4.0 seconds) using the five timestamps that occurred before them, spanning 2.0 seconds of past data. The key objective of TrajNet++ is to highlight significant agent-agent interactions across a scenario. Specifically, we evaluate the model for the subsequent 12 timestamps based on the agents' last nine timestamps. There are a total of 54,513 unique scenes in the dataset. ETH-UCY is a composite of two datasets featuring smooth trajectories and straightforward agent interactions. The ETH dataset includes two scenarios, ETH and HOTEL, totaling 750 pedestrians. On the other hand, UNIV, ZARA1, and ZARA2 scenarios, totaling 786 pedestrians, are included in the UCY dataset. These scenes encompass various settings, including roads, intersections, and open areas. The world-coordinate sequence comprises trajectories covering eight time steps or 3.2 seconds. We aim to forecast the next 12 time steps, so our predictions will cover 4.8 seconds in total.

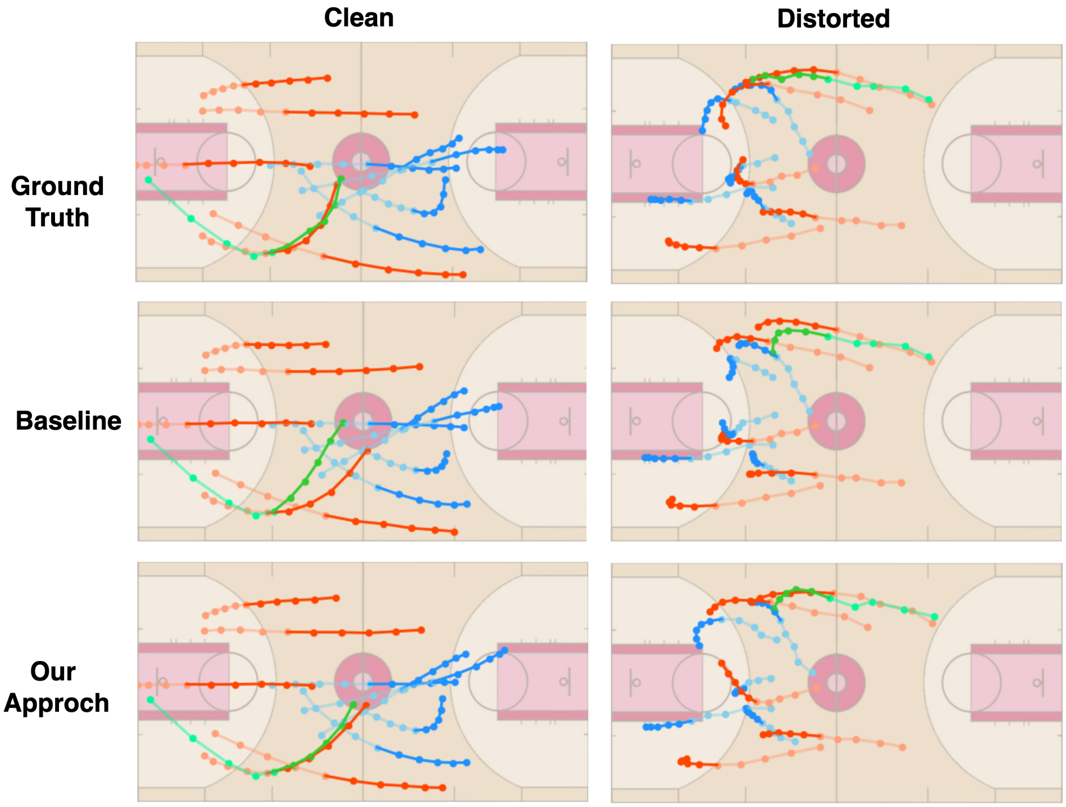

*Figure 2.* Visual representation of results on the NBA dataset. Trajectories of ten players from each team (cyan and red) are depicted alongside GroupNet (Xu et al., 2022a) and the ground truth for comparison. Past trajectories are represented in a lighter color, while predicted waypoints are shown in a solid color. The green color represents the ball trajectory. The first and second columns show the model's predictions for the next ten timestamps in both clean and distorted environments.

*Table 2.* Quantitative results using the *SSWDP* approach on the TrajNet++ dataset during training. (B) stands for the baseline model. RD(%) indicates the relative percentage difference from the baseline. The top performance is highlighted in **bold**.

| Model | Venue | ADE (↓) | FDE (↓) |
|---|---|---|---|
| Linear Extrapolation | ICLR 2022 | 0.409 | 0.897 |
| AntiSocial | ICLR 2022 | 0.316 | 0.632 |
| Ego | ICLR 2022 | 0.214 | 0.431 |
| AutoBot (B) | ICLR 2022 | 0.128 | 0.234 |
| Traj-MAE | ICCV 2023 | 0.093 | 0.181 |
| Our (B)+SSWDP | - | **0.091** | **0.162** |
| RD(%) | - | 33.8 | 36.4 |

## 4.2. Evaluation Metrics

We use standard evaluation metrics such as Average Displacement Error (ADE) and Final Displacement Error (FDE) for trajectory prediction evaluation. ADE represents the average L2 distance between predicted and ground truth trajectories across all time steps. In contrast, FDE quantifies the L2 distance at the last time step or final endpoint.

## 4.3. Quantitative Results

### 4.3.1. Evaluation on the NBA Dataset

On the NBA dataset, based on past trajectories from the last five timestamps (2.0 seconds), we forecast future trajectories for ten timestamps (4.0 seconds ahead). Table 1 summarizes the results of an evaluation involving several methods. We compare our approach with SSTGCNN (Mohamed et al., 2020), STAR (Yu et al., 2020), PECNet (Mangalam et al., 2020), NMMP (Hu et al., 2020), MemoNe (Xu et al., 2022b), NPSN (Bae et al., 2022), GroupNet (Xu et al., 2022a), MERA (Sun et al., 2023), DISTL (Cao et al., 2023), and DCG (Li et al., 2024). Our findings demonstrate a significant outperformance of our approach compared to others. Notably, at 4.0 seconds, the minimum Average Displacement Error (minADE) and minimum Final Displacement Error (minFDE) decrease to 0.90 and 1.14, respectively (with a relative improvement of 22.7% and 38.9% in ADE/FDE) compared to the baseline GroupNet (Xu et al., 2022a). Our approach shows significant improvement not

*Table 3.* ADE (↓) / FDE (↓) for trajectory prediction on the ETH-UCY dataset utilizing the *SSWDP* technique during training. (B1) and (B2) denote the first and second baseline models. RD1 (%) and RD2 (%) indicate the relative percentage difference compared to the baseline B1 and B2, respectively. The top performance is highlighted in **bold**.

| Method | Venue | ETH | HOTEL | UNIV | ZARA1 | ZARA2 | AVG |
|---|---|---|---|---|---|---|---|
| STT (Monti et al., 2022) | CVPR 2022 | 0.54/1.10 | 0.24/0.46 | 0.57/1.15 | 0.45/0.94 | 0.36/0.77 | 0.43/0.88 |
| SEEM (Wang et al., 2022a) | TPAMI 2022 | 0.48/0.86 | 0.52/1.18 | 0.35/0.65 | 0.28/0.47 | 0.24/0.50 | 0.38/0.71 |
| GroupNet (Xu et al., 2022a) | CVPR 2022 | 0.46/0.73 | 0.15/0.25 | 0.26/0.49 | 0.21/0.39 | 0.17/0.33 | 0.25/0.44 |
| DynGroupNet (Xu et al., 2023b) | NN 2023 | 0.42/0.66 | 0.13/0.20 | 0.24/0.44 | 0.19/0.34 | 0.15/0.28 | 0.23/0.38 |
| RMB (Shi et al., 2023a) | TPAMI 2023 | 0.29/0.49 | 0.12/0.18 | 0.29/0.51 | 0.20/0.36 | 0.15/0.27 | 0.21/0.36 |
| BCDiff (Li et al., 2023) | NIPS 2023 | 0.53/0.91 | 0.17/0.27 | 0.24/0.40 | 0.21/0.37 | 0.16/0.26 | 0.26/0.44 |
| EqMotion (Xu et al., 2023a) | CVPR 2023 | 0.40/0.61 | 0.12/0.18 | 0.23/0.43 | 0.18/0.32 | 0.13/0.23 | 0.21/0.35 |
| FlowChain (Maeda & Ukita, 2023) | ICCV 2023 | 0.55/0.99 | 0.20/0.35 | 0.29/0.54 | 0.22/0.40 | 0.20/0.34 | 0.29/0.52 |
| BOsample (Chen et al., 2023a) | ICCV 2023 | 0.52/0.95 | 0.19/0.39 | 0.30/0.67 | 0.14/0.33 | 0.20/0.45 | 0.27/0.56 |
| TUTR (Shi et al., 2023b) | ICCV 2023 | 0.40/0.61 | 0.11/0.18 | 0.23/0.42 | 0.18/0.34 | 0.13/0.25 | 0.21/0.36 |
| EigenTrajectory (Bae et al., 2023) | ICCV 2023 | 0.36/0.57 | 0.13/0.21 | 0.24/0.43 | 0.19/0.34 | 0.14/0.25 | 0.21/0.36 |
| Graph-TERN (B1) (Bae & Jeon, 2023) | AAAI 2023 | 0.42/0.58 | 0.14/0.23 | 0.26/0.45 | 0.21/0.37 | 0.17/0.29 | 0.24/0.38 |
| SSAGCN (B2) (Lv et al., 2023) | TNLS 2023 | 0.21/0.38 | 0.11/0.19 | 0.14/0.25 | 0.12/0.22 | 0.09/0.15 | 0.13/0.24 |
| ST-motion (Saadatnejad et al., 2023) | ICLR 2024 | 0.93/1.81 | 0.32/0.60 | 0.54/1.16 | 0.42/0.90 | 0.32/0.70 | 0.51/1.03 |
| SMEMO (Marchetti et al., 2024) | TPAMI 2024 | 0.39/0.59 | 0.14/0.20 | 0.23/0.41 | 0.19/0.32 | 0.15/0.26 | 0.22/0.35 |
| Our (B1)+SSWDP | - | 0.38/0.48 | 0.14/0.23 | 0.24/0.40 | 0.19/0.32 | 0.15/0.25 | 0.22/0.33 |
| RD1(%) ADE/FDE | - | - | - | - | - | - | 8.60/14.0 |
| Our (B2)+SSWDP | - | **0.21/0.38** | **0.08/0.10** | **0.10/0.17** | **0.11/0.19** | **0.08/0.12** | **0.11/0.19** |
| RD2(%) ADE/FDE | - | - | - | - | - | - | 16.60/23.20 |

*Table 4.* Results for the *GroupNet+SSWDP* model on the NBA dataset. *(B)* indicates the baseline GroupNet model. *(B+SC)* denotes the baseline with the spatial consistency module. *(B+SC+DP)* denotes the baseline with both the spatial consistency module and the distortion prediction module. RD(%) refers to the relative percent difference with respect to the baseline. Top performance is highlighted in **bold**.

| Method | ADE | | | | FDE | | | | ADE/FDE RD(%) |
|---|---|---|---|---|---|---|---|---|---|
| | 1.0s | 2.0s | 3.0s | 4.0s | 1.0s | 2.0s | 3.0s | 4.0s | |
| $B$ | 0.34 | 0.62 | 0.87 | 1.13 | 0.48 | 0.95 | 1.31 | 1.69 | - |
| $B + SC$ | 0.28 | 0.52 | 0.77 | 1.018 | 0.37 | 0.75 | 1.10 | 1.36 | 10.42/21.49 |
| $B + SC + DP$ | **0.23** | **0.45** | **0.67** | **0.903** | **0.31** | **0.63** | **0.92** | **1.14** | 22.33/38.28 |

*Table 5.* Results for the SSAGCN+SSWDP model on the ETH-UCY dataset. (B) indicates the baseline SSAGCN model. (B+SC) denotes the baseline with the spatial consistency module. (B+SC+DP) denotes the baseline with both the spatial consistency module and the distortion prediction module. We report the average of the minADE/minFDE of each subset within the ETH-UCY dataset.

| Method | AVG minADE | AVG minFDE |
|---|---|---|
| *B* | 0.13 | 0.24 |
| *B + SC* | 0.12 | 0.22 |
| *B + SC + DP* | 0.11 | 0.19 |

only on *short-time horizon* prediction (1-2s) but also on *longer-time horizon* prediction (2-4s).

### 4.3.2. Evaluation on the TRAJNET++ Dataset

On the synthetic partition of the TRAJNET++ Dataset, leveraging data from the preceding nine timestamps, we forecast the subsequent 12 timestamps for each agent. We compare our approach with Linear Extrapolation, AntiSocial, Ego, AutoBot (Girgis et al., 2022) and Traj-MAE (Chen et al., 2023b). The integration of *SSWDP* into the AutoBot baseline showcased enhanced performance compared to the baseline, as illustrated in Table 2. Notably, there is a substantial 33.8% improvement in ADE and a 36.4% improvement in FDE values when compared to the baseline AutoBot model.

### 4.3.3. Evaluation on the ETH-UCY Dataset

Table 3 presents a comparison of ADE and FDE values for various methods, including our approach. Notably, our approach demonstrates a significant improvement over the baselines (B1 and B2). Specifically, with the integration of SSWDP, we achieved a relative percentage gain of 8.60/14.00% concerning baseline one (B1) and 16.60/23.20% concerning baseline two (B2) in ADE/FDE. Our model (B2) + SSWDP achieved state-of-the-art results in the trajectory prediction task on the ETH-UCY Dataset, as detailed in Table 3.

*Table 6.* Results for the GroupNet+SSWDP model using various noise factor values during training. The noise factor ($\omega$) of 0.05 exhibits the best ADE/FDE values on the validation data. The top performance is highlighted in **bold**.

| Model/<br>Dataset | Noise<br>Factor | Validation Accuracy | |
|---|---|---|---|
| | | ADE | FDE |
| | | 4.0s | 4.0s |
| GroupNet,<br>NBA | 1 | 0.908 | 1.154 |
| | 0.1 | 0.905 | 1.131 |
| | 0.05 | **0.896** | **1.130** |
| | 0 | 1.13 | 1.69 |

## 4.4. Qualitative Results on NBA Dataset

We further assessed the capabilities of our approach through qualitative results. Figure 2 illustrates the predictions of our SSWDP and GroupNet in both clean and distorted environments on the NBA SportVU dataset. It is evident from Figure 2 that our approach performs better in both clean and distorted environments in most cases. This superior performance is also reflected in the ADE/FDE matrices, as demonstrated in Tables 1 and 7.

## 4.5. Qualitative Results on ETH-UCY Dataset

We have provided visualizations of predicted density on the ETH/UCY datasets as shown in Figure 3. Our approach captures the agent's future trajectory distribution by accurately predicting the future density represented by the blue color (Agent 1) and green color (Agent 2). In contrast to SSAGCN, which predicts the density slightly deviated from the ground truth, our approach precisely predicts the future density, as illustrated in Figure 3.

## 4.6. Ablation Studies

### 4.6.1. Different Data Augmentations

We tested various data augmentation approaches, including trajectory flipping, trajectory masking, and noise augmentation using GroupNet on the NBA dataset. Our findings reveal that trajectory flipping augmentation shows insignificant improvement, with only a 0.88%/2.4% relative increase in the ADE/FDE values compared to the baseline GroupNet model. Trajectory masking augmentation shows relatively better improvement, exhibiting a 4.5%/11.0% relative increase in the ADE/FDE values compared to the baseline. Noise augmentation achieves the best improvements among the three approaches, with a 10.42%/21.49% relative increase in the ADE/FDE values compared to the baseline. In contrast, our approach (distortion prediction as a self-supervised pretext task) achieves the highest improvements, demonstrating a 22.33%/38.28% relative increase in the ADE/FDE values compared to the baseline.

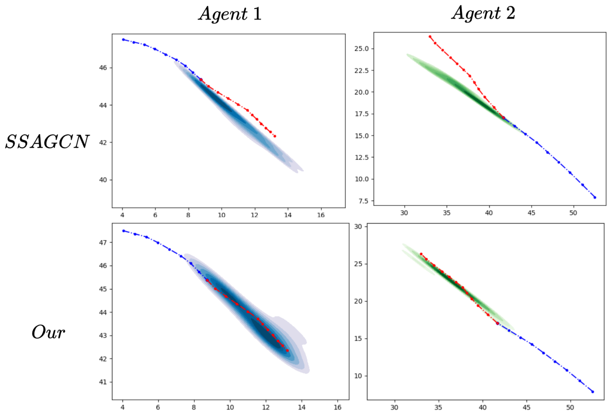

*Figure 3.* Illustration of temporal density estimations of the agent for the ETH/UCY datasets using SSAGCN (Lv et al., 2023) and our approach. The color density (blue for Agent 1 and green for Agent 2) depicts the forecasted distribution of future trajectories, with the blue dotted line representing the historical trajectory (8 timestamps) and the red dotted line corresponding to the actual ground truth (12 timestamps).

### 4.6.2. Significance of Components

We conducted experiments to validate the different modules of our approach. The results in Table 4 show that including the proposed pretext task (Model B+SC+DP) improves trajectory prediction performance from 1.13/1.69 to 0.903/1.147 (ADE/FDE values), representing a relative percentage difference of 22.33/38.28%. Furthermore, the (B+SC) model achieved a result of 1.018/1.362, which is

*Table 7.* The results from experiments, which involved introducing distorted and clean environments in the trajectory prediction task during testing, reveal that *SSWDP* demonstrates resilience, whereas baseline models experience a significant performance decline. RD(%) represents the relative percent difference compared to the baseline. The top performance is highlighted in **bold**.

| Methods | Datasets | Environment | Baseline | | Our | | RD(%) |
|---|---|---|---|---|---|---|---|
| | | | ADE | FDE | ADE | FDE | (ADE/FDE) |
| AutoBot | TrajNet++ | Clean | 0.128 | 0.234 | **0.091** | **0.162** | 33.8/36.4 |
| AutoBot | TrajNet++ | Distorted | 0.301 | 0.469 | **0.134** | **0.195** | 76.8/82.5 |
| GroupNet | NBA | Clean | 1.13 | 1.69 | **0.90** | **1.14** | 22.7/38.9 |
| GroupNet | NBA | Distorted | 1.784 | 1.771 | **0.95** | **1.23** | 61.0/36.1 |

*Table 8.* Training time and memory usage for the baseline SSAGCN model and our SSWDP model on the ETH-UCY dataset are reported. Time is measured in seconds, and memory is reported in MB. We report the training time for a single training epoch on the ETH-UCY dataset with a batch size of 128.

| Dataset | Training Time (SSAGCN) | Training Time (Our) | Training Memory (SSAGCN) | Training Memory (Our) |
|---|---|---|---|---|
| ETH | 0.98 s | 1.04 s | 656 MB | 698 MB |
| HOTEL | 3.88 s | 4.26 s | 928 MB | 970 MB |
| UNIV | 12.95 s | 13.53 s | 4372 MB | 4454 MB |
| ZARA1 | 8.52 s | 10.08 s | 920 MB | 970 MB |
| ZARA2 | 11.55 s | 12.34 s | 1314 MB | 1442 MB |

*Table 9.* Test time and memory usage for the baseline SSAGCN model and our SSWDP model on the ETH-UCY dataset are reported. During testing, we compute the inference time of the prediction model to infer the trajectory of a single agent (12 timestamps). Time is measured in seconds, and memory is reported in MB. For testing, we report the trajectory of a single agent with the batch size set to 1.

| Dataset | Test Time (SSAGCN) | Test Time (Our) | Test Memory (SSAGCN) | Test Memory (Our) |
|---|---|---|---|---|
| ETH | 0.0017 s | 0.0017 s | 460 MB | 460 MB |
| HOTEL | 0.0017 s | 0.0017 s | 468 MB | 468 MB |
| UNIV | 0.0021 s | 0.0021 s | 706 MB | 706 MB |
| ZARA1 | 0.0018 s | 0.0018 s | 576 MB | 576 MB |
| ZARA2 | 0.0018 s | 0.0018 s | 596 MB | 596 MB |

worse than the result of our (B+SC+DP) model. We also conduct similar ablation experiments on ETH-UCY as shown in Table 5.

### 4.6.3. Choice of Noise Factor

We conducted a study to evaluate the selection of the noise factor ($\omega$) for training our SSWDP. This noise factor is crucial as it regulates the impact of distortion in repositioning spatial waypoints to generate the distorted view. The results are outlined in Table 6; it is worth noting that the value of the noise factor may vary from dataset to dataset, and its determination involves the use of cross-validation.

### 4.6.4. Clean vs. Distorted Environment

We assess the effectiveness of our approach in a distorted environment by introducing distortion into past trajectories. Both the baseline model and our model use the same distorted past trajectories to make future trajectory predictions. The results are presented in Table 7. On TrajNet++, the introduction of distortion led to a deterioration in baseline method performance compared to our approach, with a relative percentage difference of 76.8/82.5% in ADE/FDE. Similarly, for the NBA dataset, we observed a difference of 61.0/36.1% in the ADE/FDE values between our model and the baseline, indicating that our model performs significantly better in a distorted environment. It is worth noting that for these experiments, we did not use the exact same amount of distortion that was used to train our model initially. This deliberate choice illustrates the generalizability of the model trained using our approach in a distorted environment.

### 4.6.5. Computational Complexity

We report the training and testing complexities in terms of memory utilization and time consumption in Tables 8 and 9. Regarding GPU utilization, our approach results in a slight increase in GPU memory usage compared to the baseline model during training. However, the GPU memory utilization and inference time during testing remain consistent with the baseline model, as only one view (clean view) is utilized.

## 5. Conclusion

This work proposes a novel approach named SSWDP (Self-Supervised Waypoint Distortion Prediction), consisting of spatial consistency and distortion prediction modules. Our approach generates clean, and distorted views of historical trajectories observed over spatial waypoints. Subsequently, we enforce the trajectory prediction model to maintain spatial consistency between predictions derived from these two views. We also propose a simple yet highly effective pretext task of distortion prediction within observed trajectories. This self-supervised pretext task contributes to a deeper understanding of underlying representations in trajectory prediction, thereby enhancing the accuracy of future predictions. Experimental results show that incorporating SSWDP into the model learning process yields substantial performance improvements, even in distorted environments, when compared to baseline methods. This underscores the potential of our approach as a valuable complement to existing trajectory prediction techniques.

## Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

## References

Bae, I. and Jeon, H.-G. A set of control points conditioned pedestrian trajectory prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6155–6165, 2023.

Bae, I., Park, J.-H., and Jeon, H.-G. Non-probability sampling network for stochastic human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6477–6487, 2022.

Bae, I., Oh, J., and Jeon, H.-G. Eigentrajectory: Low-rank descriptors for multi-modal trajectory forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10017–10029, 2023.

Bhattacharyya, P., Huang, C., and Czarnecki, K. Ssl-lanes: Self-supervised learning for motion forecasting in autonomous driving. In *Conference on Robot Learning*, pp. 1793–1805. PMLR, 2023.

Cao, C., Yang, C., and Li, S. Discovering intrinsic spatial-temporal logic rules to explain human actions. *Advances in Neural Information Processing Systems*, 2023.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 132–149, 2018.

Chen, G., Chen, Z., Fan, S., and Zhang, K. Unsupervised sampling promoting for stochastic human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17874–17884, 2023a.

Chen, H., Wang, J., Shao, K., Liu, F., Hao, J., Guan, C., Chen, G., and Heng, P.-A. Traj-mae: Masked autoencoders for trajectory prediction. *arXiv preprint arXiv:2303.06697*, 2023b.

Cheng, J., Mei, X., and Liu, M. Forecast-mae: Self-supervised pre-training for motion forecasting with masked autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8679–8689, 2023.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Girgis, R., Golemo, F., Codevilla, F., Weiss, M., D'Souza, J. A., Kahou, S. E., Heide, F., and Pal, C. Latent variable sequential set transformers for joint multi-agent motion prediction. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Dup_dDqkZC5.

Giuliari, F., Hasan, I., Cristani, M., and Galasso, F. Transformer networks for trajectory forecasting. In *2020 25th international conference on pattern recognition (ICPR)*, pp. 10335–10342. IEEE, 2021.

Gu, J., Hu, C., Zhang, T., Chen, X., Wang, Y., Wang, Y., and Zhao, H. Vip3d: End-to-end visual trajectory prediction via 3d agent queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5496–5506, 2023.

Gu, T., Chen, G., Li, J., Lin, C., Rao, Y., Zhou, J., and Lu, J. Stochastic trajectory prediction via motion indeterminacy diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17113–17122, 2022.

Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., and 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.

Halawa, M., Hellwich, O., and Bideau, P. Action-based contrastive learning for trajectory prediction. In *European Conference on Computer Vision*, pp. 143–159. Springer, 2022.

Hu, Y., Chen, S., Zhang, Y., and Gu, X. Collaborative motion prediction via neural motion message passing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6319–6328, 2020.

Kothari, P., Kreiss, S., and Alahi, A. Human trajectory forecasting in crowds: A deep learning perspective. *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2021. doi: 10.1109/TITS.2021.3069362.

Lee, M., Sohn, S. S., Moon, S., Yoon, S., Kapadia, M., and Pavlovic, V. Muse-vae: multi-scale vae for environment-aware long term trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2221–2230, 2022.

Lerner, A., Chrysanthou, Y., and Lischinski, D. Crowds by example. In *Computer graphics forum*, volume 26, pp. 655–664. Wiley Online Library, 2007.

Li, J., Hua, C., Ma, H., Park, J., Dax, V., and Kochenderfer, M. J. Multi-agent dynamic relational reasoning for social robot navigation. *arXiv preprint arXiv:2401.12275*, 2024.

Li, R., Li, C., Ren, D., Chen, G., Yuan, Y., and Wang, G. Bcdiff: Bidirectional consistent diffusion for instantaneous trajectory prediction. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Lv, P., Wang, W., Wang, Y., Zhang, Y., Xu, M., and Xu, C. Ssagcn: social soft attention graph convolution network for pedestrian trajectory prediction. *IEEE transactions on neural networks and learning systems*, 2023.

Maeda, T. and Ukita, N. Fast inference and update of probabilistic density estimation on trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9795–9805, 2023.

Mangalam, K., Girase, H., Agarwal, S., Lee, K.-H., Adeli, E., Malik, J., and 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.

Mao, W., Xu, C., Zhu, Q., Chen, S., and Wang, Y. Leapfrog diffusion model for stochastic trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5517–5526, 2023.

Marchetti, F., Becattini, F., Seidenari, L., and Del Bimbo, A. Smemo: social memory for trajectory forecasting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Mohamed, A., Qian, K., Elhoseiny, M., and Claudel, C. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14424–14432, 2020.

Monti, A., Porrello, A., Calderara, S., Coscia, P., Ballan, L., and Cucchiara, R. How many observations are enough? knowledge distillation for trajectory forecasting. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6543–6552, 2022. doi: 10.1109/CVPR52688.2022.00644.

Pathak, D., Girshick, R., Dollár, P., Darrell, T., and Hariharan, B. Learning features by watching objects move. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2701–2710, 2017.

Pellegrini, S., Ess, A., Schindler, K., and 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.

Saadatnejad, S., Bahari, M., Khorsandi, P., Saneian, M., Moosavi-Dezfooli, S.-M., and Alahi, A. Are socially-aware trajectory prediction models really socially-aware? *Transportation research part C: emerging technologies*, 141:103705, 2022.

Saadatnejad, S., Gao, Y., Messaoud, K., and Alahi, A. Social-transmotion: Promptable human trajectory prediction. *arXiv preprint arXiv:2312.16168*, 2023.

Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofighi, H., and Savarese, S. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, number CONF, 2019.

Sekhon, J. and Fleming, C. Scan: A spatial context attentive network for joint multi-agent intent prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6119–6127, 2021.

Shi, L., Wang, L., Long, C., Zhou, S., Zhou, M., Niu, Z., and Hua, G. Sgcn: Sparse graph convolution network for pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8994–9003, 2021.

Shi, L., Wang, L., Long, C., Zhou, S., Tang, W., Zheng, N., and Hua, G. Representing multimodal behaviors with mean location for pedestrian trajectory prediction. *IEEE transactions on pattern analysis and machine intelligence*, 2023a.

Shi, L., Wang, L., Zhou, S., and Hua, G. Trajectory unified transformer for pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9675–9684, 2023b.

Sun, J., Li, Y., Chai, L., and Lu, C. Modality exploration, retrieval and adaptation for trajectory prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Tsao, L.-W., Wang, Y.-K., Lin, H.-S., Shuai, H.-H., Wong, L.-K., and Cheng, W.-H. Social-ssl: Self-supervised cross-sequence representation learning based on transformers for multi-agent trajectory prediction. In *European Conference on Computer Vision*, pp. 234–250. Springer, 2022.

Wang, D., Liu, H., Wang, N., Wang, Y., Wang, H., and McLoone, S. Seem: A sequence entropy energy-based model for pedestrian trajectory all-then-one prediction. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):1070–1086, 2022a.

Wang, R., Hu, Z., Song, X., and Li, W. Trajectory distribution aware graph convolutional network for trajectory prediction considering spatio-temporal interactions and scene information. *IEEE Transactions on Knowledge and Data Engineering*, 2023a.

Wang, Y., Zhou, H., Zhang, Z., Feng, C., Lin, H., Gao, C., Tang, Y., Zhao, Z., Zhang, S., Guo, J., et al. Tenet: Transformer encoding network for effective temporal flow on motion prediction. *arXiv preprint arXiv:2207.00170*, 2022b.

Wang, Y., Zhang, P., Bai, L., and Xue, J. Fend: A future enhanced distribution-aware contrastive learning framework for long-tail trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1400–1409, 2023b.

Wei, C., Xie, L., Ren, X., Xia, Y., Su, C., Liu, J., Tian, Q., and Yuille, A. L. Iterative reorganization with weak spatial constraints: Solving arbitrary jigsaw puzzles for unsupervised representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1910–1919, 2019.

Wu, P., Majumdar, A., Stone, K., Lin, Y., Mordatch, I., Abbeel, P., and Rajeswaran, A. Masked trajectory models for prediction, representation, and control. *arXiv preprint arXiv:2305.02968*, 2023.

Xu, C., Li, M., Ni, Z., Zhang, Y., and Chen, S. Groupnet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6498–6507, June 2022a.

Xu, C., Mao, W., Zhang, W., and 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, 2022b.

Xu, C., Wei, Y., Tang, B., Yin, S., Zhang, Y., and Chen, S. Dynamic-group-aware networks for multi-agent trajectory prediction with relational reasoning. *arXiv preprint arXiv:2206.13114*, 2022c.

Xu, C., Tan, R. T., Tan, Y., Chen, S., Wang, Y. G., Wang, X., and Wang, Y. Eqmotion: Equivariant multi-agent motion prediction with invariant interaction reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1410–1420, 2023a.

Xu, C., Wei, Y., Tang, B., Yin, S., Zhang, Y., Chen, S., and Wang, Y. Dynamic-group-aware networks for multi-agent trajectory prediction with relational reasoning. *Neural Networks*, 2023b.

Xu, Y., Bazarjani, A., Chi, H.-g., Choi, C., and Fu, Y. Uncovering the missing pattern: Unified framework towards trajectory imputation and prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9632–9643, 2023c.

Ye, M., Xu, J., Xu, X., Wang, T., Cao, T., and Chen, Q. Bootstrap motion forecasting with self-consistent constraints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8504–8514, 2023.

Yu, C., Ma, X., Ren, J., Zhao, H., and Yi, S. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, pp. 507–523. Springer, 2020.

Yuan, Y., Weng, X., Ou, Y., and Kitani, K. M. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9813–9823, 2021.

Zhan, E., Zheng, S., Yue, Y., Sha, L., and Lucey, P. Generating multi-agent trajectories using programmatic weak supervision. *arXiv preprint arXiv:1803.07612*, 2018.

Zhou, Z., Wang, J., Li, Y.-H., and Huang, Y.-K. Query-centric trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17863–17873, 2023.

Zhu, Y., Ren, D., Fan, M., Qian, D., Li, X., and Xia, H. Robust trajectory forecasting for multiple intelligent agents in dynamic scene. *arXiv preprint arXiv:2005.13133*, 2020.

# A. Experimental Details

## A.1. Implementation Details

To ensure a fair comparison with the methods under consideration, we maintained their default configurations, including the trajectory sequence length and timestamps used as model input. For Autobots, we report the Scene-level minADE and Scene-level minFDE (6 samples), while for GroupNet, SSAGCN, and Graph-TERN, we report minADE/minFDE (20 samples). We selected the loss weight $\lambda$ value based on the convergence of the self-supervised loss, and one such plot for GroupNet is shown in Figure 4. We considered Gaussian noise with a mean of 0 and a standard deviation of 1 for sampling noise. The noise factor ($\omega$) is used to generate a distorted view, while $\lambda$ defines the contribution of the self-supervised loss to the total training loss. The values for the noise factor ($\omega$) and $\lambda$ used in our experimentation during the training of the model are provided in Table 10. Section 4.6.3 provides insight into the choice of the noise factor. Since we utilize two views (distorted and clean views), theoretically, the training time should double. However, as we processed both views in parallel rather than sequentially on the NVIDIA RTX A5000 GPU with AMD EPYC 7543 CPU, we observed a slight increase (approximately 1.25 times) in training time compared to the baseline model training time. Nevertheless, our approach does not impact test time since we only use one view during testing.

## A.2. Baseline Models

We assess our approach by testing it on four distinct models: a Variational Autoencoder-based model (GroupNet (Xu et al., 2022a)), Transformer-based model (AutoBot (Girgis et al., 2022)), Graph-based (SSAGCN (Lv et al., 2023)) and Goal-based model (Graph-TERN (Bae & Jeon, 2023)). GroupNet adeptly captures interactions among agents, enabling it to anticipate socially plausible trajectories through relational reasoning. When combined with a Conditional Variational Autoencoder (CVAE), GroupNet can learn complex social variables for better trajectory prediction. AutoBot is an encoder-decoder architecture utilizing transformers to construct multi-agent trajectories consistent with the scene. In this architecture, the encoder employs alternating temporal and social multi-head self-attention mechanisms to facilitate learning across time and social dimensions. The SSAGCN models the degree of influence among pedestrians using a spatial-temporal graph and forecasts trajectories that align with both social and physical feasibility. Graph-TERN captures social and temporal relationships through a pedestrian graph and then employs control point prediction to refine trajectories. Graph-TERN also overcomes accumulated errors through control points and intermediate destinations.

To demonstrate the effectiveness of our approach, we choose the best/second-best performing model in their respective categories (depending on reproducibility) for each dataset. For instance, on the ETH-UCY dataset, we select the Graph-TERN model (Bae & Jeon, 2023) as the first baseline (B1) since it is the top-performing model in the endpoint trajectory prediction category. Similarly, we select SSAGCN (Lv et al., 2023) as the second baseline (B2) because it is the best performing model in the recursive trajectory prediction category.

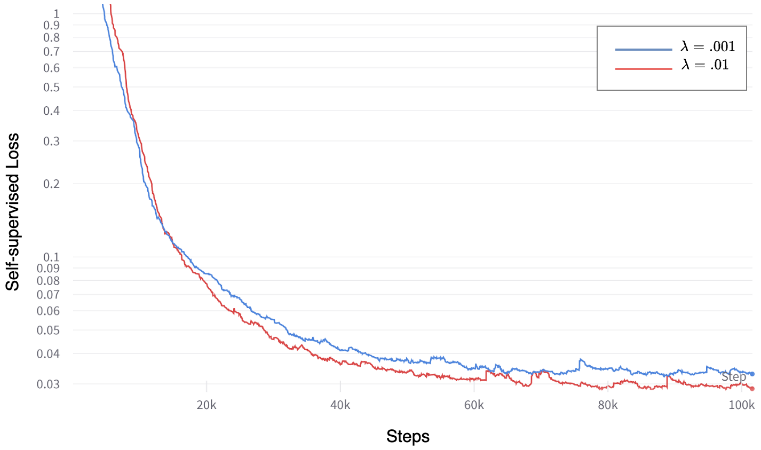

*Figure 4.* Illustration of the self-supervised loss ($\mathcal{L}_{\text{ss}}$) plot for *GroupNet+SSWDP*, indicating a decrease in loss value over the training steps on the NBA dataset. The hyperparameter value for $\lambda$ is chosen to be 0.01 (shown in red), suggesting improved learning facilitated by the distortion prediction network.

*Table 10.* Noise factor ($\omega$) and $\lambda$ values used in our experiments.

| **Dataset** | **Baselines** | $\omega$ | $\lambda$ |
|---|---|---|---|
| NBA | GroupNet | $5 * 10^{-2}$ | $10^{-2}$ |
| TrajNet | AutoBot | $10^{-1}$ | $10^{-1}$ |
| ETH, UNIV | Graph-TERN, SSAGCN | $10^{-2}$ | $10^{-1}$ |
| ZARA1, ZARA2 | Graph-TERN, SSAGCN | $10^{-1}$ | $10^{-1}$ |
| HOTEL | Graph-TERN, SSAGCN | $10^{-3}$ | $10^{-1}$ |

### A.3. Architecture Details

The SSWDP architecture comprises three primary components, as illustrated in Figure 1: the feature extractor network, the trajectory prediction network, and the distortion prediction network. The feature extraction network ($\Theta_{fe}$) generates features for both clean and distorted views. For GroupNet, $\Theta_{fe}$ represents the encoder of CVAE; for Autobot, it is the encoder of the transformer; for GraphTern, it is the multi-relational graph convolutional network; and for SSAGCN, it is a convolutional neural network. The trajectory prediction network ($\Theta_{sup}$) predicts the future trajectory. For GroupNet, the trajectory prediction network is the decoder of CVAE. For AutoBot, the trajectory prediction network is the decoder of the transformer. For GraphTern, it is the graph convolutional network. For SSAGCN, it is the temporal convolutional neural network. The distortion prediction network ($\Theta_{ss}$) predicts the distortion present in the observed past trajectory. The distortion prediction network is a simple multilayer perceptron (MLP). The input layer dimension of MLP is the dimension of output produced by the feature extraction network. The output layer dimension of MLP is the dimension of past observed trajectory. There are two hidden layers in MLP, with 128 and 64 nodes in the first and second hidden layers, respectively.