Thèse n° 9876

Deep Generative Models for Autonomous Driving: from Motion Forecasting to Realistic Image Synthesis

Présentée le 25 septembre 2023

Faculté de l'environnement naturel, architectural et construit
Intelligence Visuelle pour les Transports
Programme doctoral en génie électrique e

pour l'obtention du grade de Docteur ès Sciences

par

Saeed SAADATNEJAD

Acceptée sur proposition du jury

Prof. O. Fink, présidente du jury
Prof. A. M. Alahi, directeur de thèse
Prof. F. Nashashibi, rapporteur
Dr L. Palmieri, rapporteur
Dr M. Salzmann, rapporteur

École
polytechnique
fédérale
de Lausanne

2023

To my mother, father, and wife

Acknowledgements

The journey of this work has been made possible through the unwavering support of numerous individuals. Foremost, my gratitude extends to my supervisor, Alexandre Alahi, whose constant guidance and invaluable feedback have been pivotal in shaping this work. His consistent availability for research discussions has been noteworthy, and his encouragement to explore novel research ideas has served as a deep well of inspiration.

I express my appreciation to the esteemed members of my thesis committee – Olga Fink, Mathieu Salzmann, Fawzi Nashashibi, and Luigi Palmieri – for their thorough review of this dissertation and their insightful discussions that have enriched its content.

This thesis has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 754354.

Throughout my journey, I have been fortunate to have the support of numerous friends and collaborators at the VITA lab. I would like to begin by thanking Mohammadhossein Bahari. His friendship has not only enriched my academic pursuits but has also been a source of constant inspiration and strength during many challenging moments. I am grateful to the entire lab, especially Parth Kothari, Yuejiang Liu, Lorenzo Bertoni, and George Adaimi. The moments we have shared, whether on the campus or outside it, have been truly memorable. Conversations with Brian Sifringer, Kirell Benzi, and Bastien Van Delft have been enjoyable. Taylor Mordan's keen eye for detail and our enlightening discussions have been deeply appreciated as well. I express my sincere gratitude to our lab's exceptional secretaries during my PhD, Laurence Fonjallaz and Carol Ortega, for their expert administrative handling and kindness.

A special acknowledgment goes to Seyed-Mohsen Moosavi-Dezfooli; I have gained invaluable insights into research from him, and collaborating with him has been an enriching experience.

Through my academic journey, I had the honor of mentoring several students, each contributing to my growth. Smail Ait Bouhsain, Ahmad Rahimi, Yi Zhou Ju and Yang Gao have displayed impressive dedication, and I am proud of their journey as aspiring scholars.

During my PhD, I had the privilege of being part of the EPFLInnovators program, an experience that introduced me to remarkable individuals who ignited and nurtured my entrepreneurial aspirations. I wish all my peers within the program, great success in their future endeavors.

Lastly, the bedrock of my journey has been the unflinching support of my family. To my brothers, Vahid and MohammadAmin, and my grandmother, I have always missed being there with them,

and I profoundly appreciate all the backing they have provided.
I wish to express my heartfelt appreciation to my parents, Nahid Sarafraz and Hossein Saadatnejad, for their boundless love, unwavering belief, and immeasurable sacrifices that have made this journey possible. Their presence, even from afar, has been my guiding light.
In closing, I pay tribute to my wife, Seyedeh Fatemeh Zaker, for her enduring patience, invaluable support, and constant encouragement that have been my driving force.

Lausanne, 31 July 2023 Saeed Saadatnejad

Abstract

Forecasting is a capability inherent in humans when navigating. Humans routinely plan their paths, considering the potential future movements of those around them. Similarly, to achieve comparable sophistication and safety, autonomous systems must embrace this predictive nature. Deep generative models have played a pivotal role in advancing autonomous driving in recent years. These models are not only used in forecasting trajectory (coarse-grained) and human pose (fine-grained) but also in generating realistic synthetic images. These synthetic images, presenting intricate and diverse scenarios, provide a rigorous testing ground for evaluating the efficacy of our forecasting models.

The thesis begins with generative models in trajectory forecasting. We present a novel automated assessment, an essential but as unexplored approach, to objectively evaluate the performance of forecasting models. Our proposed adversarial generation serves as an alternative for extensive real-world testing, shedding light on how state-of-the-art models can generate forecasts that violate social norms and scene constraints. Furthermore, we leverage adversarial training to enhance model robustness against adversarial attacks and improve social awareness and scene understanding. As the thesis progresses, we delve into the impact of additional visual cues that humans subconsciously exhibit when navigating space. We present a universal approach that employs the power of transformers to effectively manage diverse available visual inputs. Drawing inspiration from prompts in natural language processing, this method demonstrates improved accuracy in human trajectory forecasting by augmenting input trajectory data.

Moving on to a fine-grained representation, pose forecasting, we first contribute an open-source library that includes various models, datasets, and standardized evaluation metrics, with the aim of promoting research and moving toward a unified and fair evaluation. Subsequently, we address the crucial but neglected aspect of uncertainty in forecasting. In an attempt to enhance model performance and trust, we introduce methods for incorporating prior knowledge about the uncertainty pattern in time and for quantifying uncertainty through clustering and entropy measures. In the face of real-world noisy observations, we propose a generic diffusion-based approach for pose forecasting. By framing the task as a denoising problem, our method presents significant improvement over state-of-the-art techniques across multiple datasets, under both clean and noisy conditions.

Finally, the thesis journeys into the realm of realistic image synthesis, offering a semantically-

aware discriminator that enriches the training of conditional generative adversarial networks. This approach enhances the traditional task of the discriminator, leading to more realistic and semantically rich image generation, thus proving useful in autonomous driving simulators.
In the spirit of open-source innovation, this thesis contributes to the collective knowledge in the field of computer vision, robotics and transportation by publicly sharing our forecasting library, along with the source code and models of our work.

Key words: Autonomous Driving, Motion Forecasting, Deep Generative Models, Human Pose Prediction, Human Trajectory Prediction, Adversarial Attack, Diffusion Models, Transformers, Generative Adversarial Networks, Image Synthesis.

Résumé

L'anticipation est une capacité inhérente à l'être humain lorsqu'il navigue l'espace. Les humains planifient régulièrement leurs trajectoires en tenant compte des mouvements futurs potentiels des personnes qui les entourent. De même, pour atteindre un niveau de sophistication et de sécurité comparable, les systèmes autonomes doivent intégrer cette nature prédictive. Les modèles génératifs profonds ont joué un rôle essentiel dans l'évolution de la conduite autonome au cours des dernières années. Ces modèles ne sont pas seulement utilisés pour prévoir la trajectoire et la pose humaine, mais aussi pour générer des images synthétiques réalistes. Ces images synthétiques, qui présentent des scénarios complexes et variés, constituent un terrain d'essai rigoureux pour évaluer l'efficacité de nos modèles de prédiction.

La thèse commence par les modèles génératifs pour la prédiction de trajectoires. Nous présentons une nouvelle méthode pour évaluer objectivement la performance des modèles de prédiction. Notre proposition de génération "adversarial" sert d'alternative pour des tests approfondis dans le monde réel, mettant en lumière la façon dont les modèles de pointe peuvent générer des prédictions qui violent les normes sociales et les contraintes de la scène. En outre, nous tirons parti de l'entraînement "adversarial" pour renforcer la robustesse du modèle contre les attaques "adversarial" et améliorer la modelisation sociale et la compréhension de la scène. Au fur et à mesure que la thèse progresse, nous étudions l'impact des indices visuels que les humains présentent inconsciemment lorsqu'ils naviguent dans l'espace. Nous présentons une approche universelle qui utilise la puissance des "Transformer" pour gérer efficacement les diverses entrées visuelles disponibles. S'inspirant des "promts" dans le traitement du langage naturel, cette méthode démontre une meilleure précision dans la prédiction de la trajectoire humaine en augmentant les données d'entrée de la trajectoire.

Passant à une représentation plus fine, la prédiction de la pose, nous contribuons d'abord à une librairie qui comprend divers modèles, des données et des mesures d'évaluation, dans le but de promouvoir la recherche et de progresser vers une évaluation unifiée et équitable. Ensuite, nous abordons l'aspect crucial mais négligé de l'incertitude dans les prédictions. Afin d'améliorer les performances des modèles et la confiance qu'ils inspirent, nous introduisons des méthodes permettant d'intégrer des connaissances préalables sur le modèle d'incertitude dans le temps et de quantifier l'incertitude au moyen de mesures de regroupement et d'entropie. Face aux observations bruitées du monde réel, nous proposons une approche générique basée sur la diffusion pour la

prédiction de la pose. En présentant la tâche comme un problème de débruitage, notre méthode présente une amélioration significative dans des conditions bruyantes.
Enfin, la thèse aborde le domaine de la synthèse d'images réalistes, en proposant un discriminateur qui enrichit l'apprentissage des réseaux "adversarial" génératifs conditionnels. Cette approche améliore la tâche traditionnelle du discriminateur, conduisant à une génération d'images plus réalistes et sémantiquement plus riches, s'avérant ainsi utile dans les simulateurs de conduite autonome.
Dans l'esprit de l'open-source, cette thèse contribue aux domaines de la vision par ordinateur, de la robotique et du transport en partageant publiquement notre librairie de prédiction, ainsi que le code source et les modèles de notre travail.

Mots clés : Conduite autonome, prédiction de mouvement, modèles génératifs profonds, prédiction de la pose humaine, prédiction de la trajectoire humaine, attaque adversariale, modèles de diffusion, réseaux adversariaux génératifs, synthèse d'images.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

In the dawn of a new era, our world is being consistently reshaped by the expanding applications of artificial intelligence (AI). Among these, the potential of AI to transform mobility is noteworthy, encompassing not only autonomous driving [1]–[3] but also socially-aware robotics [4] and delivery robots [5]. According to the Global Status Report on road safety [6], approximately 1.35 million people die each year on the roads. The deployment of autonomous vehicles (AV) promises to reduce significantly road accidents attributed to human error as well as improve the lives of individuals, providing enhanced accessibility for people with disabilities, optimizing goods delivery. This positions autonomous vehicles at the forefront of both academic and industrial interests. Nevertheless, the safety-critical nature of these systems also presents considerable challenges.

One of the most significant challenges for autonomous driving lies in accurately predicting the motion of surrounding agents, both in simulations and real-world tests. Autonomous agents must perceive their surroundings and also predict how these dynamics might evolve. These predictions can be examined at different granularity levels, from the coarse-grained forecasting of trajectories to the fine-grained forecasting of human poses.

Generative models, with their proficiency in learning data distributions, have excelled in numerous AI applications such as computer vision [7], [8], natural language processing [9], and robotics [10]. Lately, their potential use in autonomous driving tasks, from image synthesis to motion forecasting, is gaining interest. The task ahead is to harness these models effectively to enhance the safety and reliability of autonomous systems, paving the way toward wider adoption of autonomous driving.

1.2 Problems

This thesis is oriented toward advancing safer autonomous driving by focusing on two crucial problem domains: 1) addressing a variety of challenges that arise in motion forecasting and refining the use of generative models in their training, and 2) identifying issues in image synthesis with generative models, a critical component in constructing realistic simulations for training and testing autonomous driving systems.

Forecasting the future is often considered an essential aspect of intelligence [11]. This capability becomes critical in autonomous vehicles, where accurate predictions can help avoid accidents involving humans. For instance, consider a scenario where a pedestrian is about to cross the street. A non-predictive agent may only detect the pedestrian when they are directly in front, only attempting to avoid a collision at the last moment. In contrast, a predictive agent can anticipate the pedestrian's actions several seconds ahead of time, making informed decisions as to when to stop or proceed.

To assist in the definition of motion forecasting, it is essential to first differentiate between the concepts of trajectory and pose. A trajectory refers to the temporal progression of coarse-grained motion states—specifically, the position and velocity of a moving agent, such as a vehicle or a person. Yet, for certain applications, there is a need to predict[1] more fine-grained details, such as human pose keypoints. These distinct forms of motion can be seen in Figure 1.1. We define trajectory forecasting as follows:

Forecasting future human trajectories, given a sequence of past observed ones.

While other features may optionally be included as input based on their availability, the trajectory is the essential input. Similarly, we can define human pose forecasting as:

Forecasting a sequence of future human poses, given a sequence of past observed ones.

In this setting, "human pose" refers to the spatial arrangement of specific points, known as keypoints, on the human body. These keypoints can help infer the body's overall position and movement. As an illustration, the COCO dataset defines 17 such keypoints for capturing human poses [12].

In what follows, we delve into a comprehensive discussion of the problem areas that we focus on, encompassing three main aspects: (1) trajectory forecasting, (2) human pose forecasting, and (3) image synthesis intended for simulation applications.

[1] In the scope of this thesis, we use the terms "prediction" and "forecasting" synonymously and interchangeably.

Figure 1.1: The motion forecasting task: Trajectory forecasting as coarse-grained motion (left) and human pose forecasting as fine-grained motion (right). The background image is sourced from the JAAD dataset [13].

1.2.1 Trajectory Forecasting

The exploration begins with a focus on coarse-grained motion forecasting, specifically trajectory forecasting. This task can be framed as a sequence prediction problem where the goal is to anticipate future states of pedestrians based on their past states. It has inherent dependency on surrounding agents and scene configuration. Thus, an integral challenge within trajectory forecasting lies in effectively modeling social interactions and scene-awareness. In particular, a proficient forecasting model aims to capture the dependencies, in the form of social interactions, among several interrelated sequences, such as pedestrian trajectories. It is also expected to adhere to the constraints of the scene, by avoiding the prediction of movements in areas deemed impossible due to the limitations set by the scene's structure.

Deep generative models have made significant advancements in those aspects in recent years [14], [15]. The critical inquiry we must address is: do these models genuinely learn all expected aspects from the data they were trained on? The safety-sensitive nature of trajectory forecasting necessitates meticulous evaluation of prediction methods to minimize potential risks involving humans. In this context, two crucial factors must be considered: the models' social-awareness, *i.e.*, their ability to predict trajectories without causing collisions with other agents, and their scene-awareness, implying their consideration of the surrounding environment. These aspects have been duly acknowledged as key components in the field. However, what remains underexplored is the rigorous evaluation of the robustness of these models and a comprehensive and fair assessment of their overall performance. Traditionally, autonomous vehicle performance is validated through extensive real-world testing under a variety of challenging conditions. Yet, collecting and annotating data for all possible real-world scenes is neither practical nor economical. To address this issue, we propose a novel automated assessment, an essential but as unexplored approach, to objectively evaluate the performance of forecasting models. Our findings reveal a significant discrepancy: the state-of-the-art forecasting models often fail to meet our assumptions regarding their social interactions and scene-awareness.

Having studied trajectory forecasting models that take into account scene considerations and

interaction with others, we raise the following question: is it possible to develop a universal model that assimilates all available visual cues? Traditional predictors typically rely on a single data point per individual, i.e., their x-y coordinates on the ground plane. This approach, however, overlooks a rich array of additional signals—body language, social interactions, gaze directions—that humans use as visual cues to indicate intended trajectories. We will propose a generic model that harnesses these visual cues that humans subconsciously emit when navigating space.

1.2.2 Human Pose Forecasting

The relatively new field of human pose forecasting has attracted significant attention due to its vital role in areas like social robotics and autonomous navigation, assistive robotics, and human-robot interaction. While it expands upon the principles of trajectory forecasting, it delves into greater detail, enhancing our understanding of motion and behavior. The field is advancing at a quick pace. However, this happens at the cost of unfair and non-unified evaluations, as different studies employ varying metrics and dataset setups, leading to some inconsistencies between reported numbers. As a countermeasure, we develop and publicly release an open-source library to standardize the implementation and evaluation process in human pose forecasting, encouraging cohesive and collaborative progress in this research area.

The undertaking of forecasting human poses is intricate and fraught with challenges, necessitating a complex blend of spatial and temporal reasoning. It is further complicated by the broad range of scenarios and the inherent unpredictability of human behavior. The elevated level of uncertainty in this field can make it daunting, as precise forecasting of future human movements can elude even human actors. It is important for an autonomous agent to not only forecast human movements, but also to identify situations characterized by uncertainty and respond appropriately. In this thesis, two distinct types of uncertainty are addressed to enhance performance and engender a greater degree of trust in the forecasts.

The process of forecasting human poses in real-world scenarios inevitably contends with noisy inputs. Existing models have achieved satisfactory results under ideal conditions, but their performance significantly degrades in observations fraught with noise. Factors such as minor offsets in detection methods or partial occlusions of body parts can profoundly undermine prediction accuracy. In response to this challenge, we propose a novel deep generative model capable of delivering reliable results in both noiseless and noisy observations.

1.2.3 Image Synthesis for Simulators

Before deploying autonomous vehicles in real-world scenarios, it is crucial to validate their performance and ensure their reliability. Relying solely on real-world testing for the evaluation of these systems is impractical, time-consuming, and potentially unsafe. To overcome these limitations, simulations play a vital role. By leveraging generative models, such as those used in

Figure 1.2: Self-driving cars should react well in rare scenarios, such as adverse weather conditions (left) and encountering unexpected obstacles like a child running in front of the car (right). A trustable simulator is needed to ensure safety. Images were generated using Stable Diffusion [7].

image synthesis, simulations offer a controlled and cost-effective environment for assessing the capabilities of motion forecasting models. Simulations enable the exploration of diverse scenarios, including rare cases and challenging conditions that are difficult to replicate in real-world settings (two examples shown in Figure 1.2). Additionally, by synthesizing images, simulations facilitate not only the evaluation of AV performance but also the improvement of current deep networks by leveraging abundant data.

In this context, our focus lies on the task of semantically-driven image synthesis. Given a semantic input, such as human body poses or scene segmentation maps, our goal is to generate realistic images that faithfully represent the provided semantics. Realistic image synthesis is a challenging task due to the high dimensionality of the output space and the ill-posed nature of the objective. Existing models often lack detailed object representation and fail to achieve sufficient photorealism, as they prioritize high-level object structure. We aim to explore techniques to learn representations that enable accurate supervision of deep generative models to obtain high-fidelity images.

Figure 1.3 provides a visual summary of the tasks that this thesis addresses.

1.3 Thesis Contributions

Socially-aware trajectory forecasting: We propose an adversarial attack to assess the social understanding of trajectory forecasting models in terms of collision avoidance. Technically, we define collision as a failure mode of the output, and propose hard- and soft-attention mechanisms to guide our attack in this multimodal regression task. Adversarial training using our approach can not only make those models more robust against adversarial attacks, but also improve their

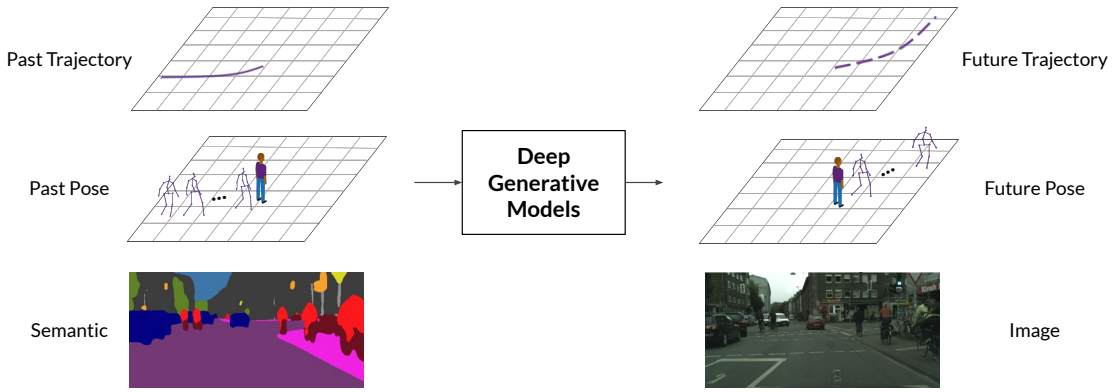


Figure 1.3: Graphical illustration of the key tasks covered in this thesis: trajectory and pose forecasting, and image synthesis.

collision avoidance. Thanks to our technique, we shed light on the common weaknesses of trajectory forecasting models, opening a window to their social understandings. The work in this chapter expands on our paper published in TR_C'22 [16]. [Details in: Chapter 2]

Scene-aware trajectory forecasting: We frame the problem through the lens of adversarial scene generation and present a method that automatically generates realistic scenes causing state-of-the-art models to go off-road. The method is a simple yet effective generative model based on atomic scene generation functions along with physical constraints. Our experiments show that more than 60% of existing scenes from the current benchmarks can be modified in a way to make prediction methods fail (*i.e.,* predicting off-road). Furthermore, we show that the generated scenes are realistic, as they exist in the real world, and can be used to enhance the robustness of existing models, resulting in a reduction of $30 - 40\%$ in the off-road rate. The work presented in this chapter is based on our paper presented at CVPR'22 [17]. [Details in: Chapter 3]

Trajectory forecasting using visual cues: We present a novel approach that leverages transformers to effectively handle various visual cues and capture the diverse and multi-modal nature of human behavior. Our approach integrates the sequence of observed cues, such as x-y coordinates, bounding boxes, or body poses, with the observed trajectories to predict future trajectories. Drawing inspiration from the concept of prompts in natural language processing (NLP), we apply the notion of prompts to human trajectory forecasting, augmenting the trajectory data and ultimately improving the accuracy of human trajectory forecasts. This chapter draws from our paper under review [18]. [Details in: Chapter 4]

Pose forecasting library including uncertainty: To foster further research and standardize evaluation, we first lay the groundwork by building an open-source library for human pose forecasting. This comprehensive library encompasses multiple models, various datasets, and standardized evaluation metrics. Building on this, we then delve into the critical aspect of

uncertainty, adding it to the pose forecasting objective to increase performance and convey better trust: 1) we propose a method for modeling inherent uncertainty by using uncertainty priors to inject knowledge about the pattern of uncertainty. This focuses the capacity of the model toward more meaningful supervision while reducing the number of learned parameters and improving stability; 2) we introduce a novel approach for quantifying the uncertainty coming from the model's lack of knowledge. To this end, forecast poses are clustered and the entropy of their assignments is measured. Our experiments demonstrate up to 25% improvements in accuracy and better performance in uncertainty estimation. This chapter is based on our paper under review [19]. [Details in: Chapter 5]

Pose forecasting in noisy observations: We propose a a diffusion-based method aimed at enhancing prediction accuracy in the presence of noisy observations. This is achieved by redefining the prediction task as a denoising challenge. In our model, we consider both observation and prediction as components of a singular sequence, with missing elements distributed throughout the sequence. We interpret these missing elements, whether in the observation or prediction horizon, as noise and denoise them with our conditional diffusion model. We acknowledge that long-term forecasting can bring added complexity so we put forward a two-tier cascaded diffusion model. In this structure, the first model focuses on short-term predictions, effectively setting a foundation, while the second one builds upon this to forecast over longer horizons. The effectiveness of our model is validated by its performance on multiple datasets, where it consistently outperforms current state-of-the-art methods. A significant advantage of our framework is its generic nature, allowing it to enhance any pose forecasting model. It can be applied as both a pre-processing step to repair faulty inputs and as a post-processing step to refine model outputs, ensuring robust and reliable predictions. The work in this chapter expands on the paper we presented at ICRA'23 [20]. [Details in: Chapter 6]

Realistic image synthesis: Our principal offering lies in the introduction of a novel semantically-aware discriminator, engineered to furnish accurate supervisions for the generator of Generative Adversarial Networks via a multi-task learning approach. Our discriminator consists of three heads: the semantics head ensuring the generator's fidelity to the semantics, the coarse-to-fine adversarial head preserving fine-grained details in the generated image, and the reconstruction head acting as a training regularizer. This alignment strengthens the consistency of the output and enhances training stability and image detail. Our contributions are generic and applicable to any generator network for image synthesis. The work in this chapter encapsulates our paper published in T-ITS'21 [21]. [Details in: Chapter 7]

1.4 Thesis Structure

The thesis is structured into eight chapters, beginning with the introduction.

As highlighted in Table 1.1, our journey begins with an evaluation of trajectory forecasting models in Chapter 2 and Chapter 3, shedding light on their strengths and vulnerabilities. Transitioning

into Chapter 4, we propose a universal trajectory forecasting model that leverages all available visual cues. The narrative then continues into the area of pose forecasting. In Chapter 5, the development of a comprehensive open-source library is presented, followed by an examination of the dual types of uncertainty intrinsic to pose forecasting. In Chapter 6, a generic diffusion-based pose forecasting model is introduced, showcasing its resilience and adaptability in both noiseless and noisy observations. We then journey into the realm of realistic image synthesis through generative models in Chapter 7.

Lastly, in Chapter 8, we summarize the contributions made in this thesis and outline potential avenues for future research. It is important to note that instead of a separate chapter dedicated to related work, each chapter builds upon the related works specific to its topic.

Chapter	**Focus**	**Output Domain**
Chapter 1	Introduction	
Chapter 2	Evaluating socially-aware trajectory forecasting	Trajectory
Chapter 3	Evaluating scene-aware trajectory forecasting	Trajectory
Chapter 4	A universal trajectory forecasting using visual cues	Trajectory
Chapter 5	Pose forecasting library including uncertainty	Pose
Chapter 6	Pose forecasting in noisy observations	Pose
Chapter 7	Realistic image synthesis for simulators	Image
Chapter 8	Summary and future works	

Table 1.1: Summary of thesis chapters.

1.5 Related Publications

This thesis is based on the material published in the following papers:

1. S. Saadatnejad, M. Bahari, P. Khorsandi, M. Saneian, S. Moosavi-Dezfooli and A. Alahi, *Are socially-aware trajectory prediction models really socially-aware?*, Transportation Research Part C: Emerging Technologies (TR_C), 2022

2. M. Bahari*, S. Saadatnejad*, A. Rahimi, M. Shaverdikondori, S. Moosavi-Dezfooli and A. Alahi, *Vehicle trajectory prediction works, but not everywhere*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022

3. S. Saadatnejad, S. Li, T. Mordan, and A. Alahi, *A Shared Representation for Photorealistic Driving Simulators*, IEEE Transactions on Intelligent Transportation Systems (T-ITS), 2021

4. S. Saadatnejad, A. Rasekh, M. Mofayezi, Y. Medghalchi, S. Rajabzadeh, T. Mordan and A. Alahi, *A generic diffusion-based approach for 3D human pose prediction in the wild*, IEEE International Conference on Robotics and Automation (ICRA), 2023

5. S. Saadatnejad, M. Mirmohammadi, M. Daghyani, P. Saremi, Y. Zoroofchi Benisi, A. Alimohammadi, Z. Tehraninasab, T. Mordan, A. Alahi, *Toward Reliable Human Pose Forecasting with Uncertainty*, under review, 2023

6. S. Saadatnejad, Y. Gao, K. Messaoud and A. Alahi, *Social-Transmotion: Promptable Human Trajectory Prediction*, under review, 2023

7. S. Saadatnejad, B. Parsaeifard, Y. Liu, T. Mordan, and A. Alahi, *Learning Decoupled Representations for Human Pose Forecasting*, Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, 2021

8. S. Saadatnejad, S.A. Bouhsain, and A. Alahi, *Pedestrian Intention Prediction: A Multi-task Perspective*, hEART, 2020

2 Socially-aware Trajectory Forecasting

This chapter is based on the article:

Saeed Saadatnejad, Mohammadhossein Bahari, Pedram Khorsandi, Mohammad Saneian, Seyed-Mohsen Moosavi-Dezfooli and Alexandre Alahi, *Are socially-aware trajectory forecasting models really socially-aware?*, Transportation Research Part C: Emerging Technologies (TR_C), 2022

The code and a summary video of the work can be found on the project's webpage[1].

2.1 Introduction

Understanding the social behavior of humans is a core problem in many transportation applications ranging from autonomous navigation (*e.g.*, social/delivery robots [5] or autonomous vehicles [22]), to microscopic pedestrian flow simulations [23], [24]. For a robot to navigate among crowds safely or for an autonomous vehicle to drive in urban areas harmlessly, human behavior anticipation is essential. In particular, dealing with humans makes the problem safety-critical. For instance, a self-driving car's wrong prediction in a crosswalk can put a pedestrian's life in danger. Being a safety-critical problem raises the need for careful assessments of the trajectory forecasting methods to mitigate the risks associated with humans. Consequently, the robustness properties of those methods, as one of the important assessment aspects, should be carefully studied.

The pedestrian trajectory forecasting problem is to predict future positions of pedestrians given their past positions as inputs. The problem has received solutions based on microscopic human simulation models [26], [27]. Recently, diverse approaches based on neural networks are proposed. Various models based on Long-Short-Term-Memory networks [25], convolutional neural networks [28], and Generative Adversarial Networks (GAN) [29] are presented. The main challenge of the problem lies in learning the interactions between people. Therefore, socially-

[1] https://s-attack.github.io/

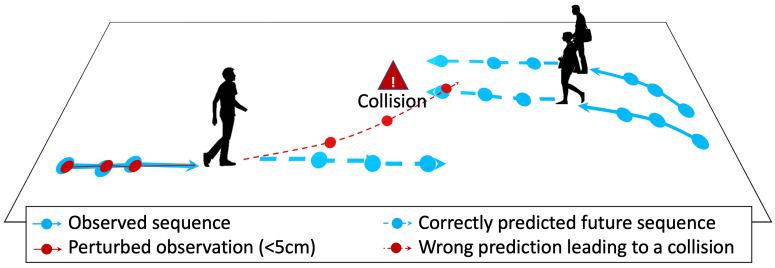


Figure 2.1: Given the observed trajectories of the agents in the scene, a predictor (here S-LSTM [25]) forecasts the future positions reasonably (blue lines). However, with less than 5 cm perturbation in the observation trajectory (in red), an unacceptable collision is predicted.

aware neural network-based models are designed to tackle the interaction more accurately [25], [30]–[32]. Human interactions involve different social behaviors such as collision avoidance, walking within a group, and merging from different directions into a specific point. Among all behaviors, collision avoidance, *i.e.,* people choosing a path that avoids collision with others, is one of the key behaviors rarely violated. That is why many previous works consider respecting collision avoidance as the evidence of their model being social [14], [31], [32]. Similarly, we consider collision avoidance as an indicator of social behavior of the models.

We show a conceptually plausible real-world scenario in Figure 2.1. Given the observed trajectories of humans in the scene, a social predictor forecasts the future positions reasonably without collision. However, by adding a small perturbation of less than 5 cm to the observation trajectory, unexpectedly, a collision between predictions of agents occurs which indicates a non-complete social understanding by the predictors. The trajectories in that figure comes from S-LSTM [25].

In this work, in contrast to the common adversarial attacks which are designed for classifiers [33], [34], we design attack for the trajectory forecasting problem which is a multimodal regression task. We use adversarial examples to study the collision avoidance behavior of the trajectory forecasting models. More specifically, we investigate the worst-case social behavior of a prediction model under small perturbations of the inputs. This study has two primary motivations; (1) it is an evaluation method for the previously-proposed predictors. Our method brings counter-examples in which the models fail in having social behavior, *i.e.,* it cannot avoid collision. (2) leveraging adversarial examples, one can train models with better collision-avoidance. Furthermore, our study highlights practical concerns for employing such models in real-world applications. Notably, it is shown that state-of-the-art localization algorithms give on-average more than 0.2 m errors on human location detection at each frame [35], [36]. While our work focuses on model failures under adversarial settings, it motivates further studies of the model's performance when localization algorithms' error distributions are concerned.

We propose an adversarial attack to fool/fail the trajectory forecasting models by causing collision between two agents' *predicted trajectories*. Namely, the adversarial attack aims at finding small

perturbations that lead to a collision. The collision can hypothetically happen between any two agents and at any prediction timestep. However, from the attack algorithm's perspective, the choice of the agent and the timestep impacts the final perturbation's size significantly. To address that, we introduce an attention-guided adversarial attack, named *Socially-ATTended ATTack (S-ATTack)*, which learns the best collision points. We present two variants of our attack: hard-attention and soft-attention. Our experiments demonstrate that our novel attack can find perturbations that make state-of-the-art trajectory forecasting models generate wrong predictions, leading to collisions with small perturbations. We show that the achieved perturbations for one predictor can be transferred to other predictors revealing common weaknesses among models. Also, we demonstrate that the models are over-dependant on the last observation points which makes the models vulnerable. Lastly, we introduce an adversarial training scheme to make trajectory forecasting models more robust. In particular, we show how our method can improve the models' social understanding in terms of collision avoidance. To the best of our knowledge, this is the first work addressing the adversarial vulnerability and robustness of trajectory forecasting models. Our main contributions are summarized as follows:

- We introduce S-ATTack to assess the social understanding of the state-of-the-art trajectory forecasting models.
- Our experiments shed light on the weaknesses of prediction models from different aspects.
- We demonstrate how to improve the robustness properties of the predictors using our S-ATTack.

2.2 Related Work

2.2.1 Human Trajectory Forecasting

Social-force model [37] was one of the key hand-crafted methods proposed to capture human social behaviors. They model interactions between pedestrians by means of social fields determined by repulsive and attractive forces. Social interactions have been addressed from other perspectives such as discrete choice modeling [38], and cellular automaton model [39], [40]. While all these hand-crafted methods have nice interpretability and are not data-heavy, it is shown that they are not able to effectively model long-term dependencies or complex interactions [41].

Social-LSTM [25] was the first work that proposed the use of data-driven neural networks instead of hand-crafted functions to learn interaction dynamics between agents in the scene. Many works pursued the use of deep learning and proposed diverse approaches for learning interactions. Different types of pooling social features are studied to share features of agents leading to a social-compliant prediction [14], [25], [42], [43]. Convolutional neural networks were studied to process pooled features of agents in the scene [28]. In order to better detect relative importance of each agent, attention networks were employed [30]. Recently many works leveraged graph neural networks to model relations between agents [31], [32], [44], [45]. The multimodal distribution of

data was learned by using generative adversarial networks [29], [46], [47]. While all mentioned works improved the performance on average and final displacement errors, they occasionally output unacceptable solutions (e.g., collisions). In this work, we attack the state-of-the-art deep learning-based recent models and reveal their weaknesses against small perturbations which challenges their social behavior.

Previously mentioned works try to learn social behavior by observing human data, which implicitly include collision avoidance. There also exist works that explicitly teach models to avoid collisions. While most of these works address the planning problem [48], [49], some adopted the inverse reinforcement learning framework to guide the network toward collision-free trajectories [50]. In this work, we focus our attack on the deep learning-based trajectory predictors which are fully data-driven and do not leverage human guidance.

2.2.2 Adversarial Attacks

Adversarial attack was first introduced by showing how vulnerable deep image classifiers are against non-perceptible yet carefully-crafted perturbations [34]. Formally, an adversarial perturbation is defined as the minimal perturbation R which changes the output of the given classifier f:

$$\min_{R} ||R||_2 \text{ subject to } f(X+R) \neq f(X), \tag{2.1}$$

where X is the input image. It was shown that adding these imperceptible perturbations bounded in terms of ℓ_p norms can easily fool image classifiers [51]–[53]. According to the "human vision knowledge", adding imperceptible perturbations to an image does not change its category label. Therefore, a good model should preserve its prediction while adding these perturbations to its input.

Adversarial attacks are also generalized to assess vulnerability of models against perturbations in other domains such as natural language processing [54], [55] and tabular data [56]. Nevertheless, to the best of our knowledge, adversarial attacks were not yet explored in the context of human trajectory forecasting. Human trajectory is a temporal trajectory, hence can be seen as time-series data prediction. Different types of time-series data were studied in classification problems [57]–[59]. While their focus was on the classification task, we target a regression task which makes the definition of a wrong output challenging. In addition, in human trajectory sequences, the imperceptibility of the perturbations is not the main interest because here, human knowledge is to respect social behaviors (avoiding collisions) in all scenarios. Lastly, a specific challenge to our problem is the freedom in the choice of the collision point which affects the perturbation size. We address these challenges by the proposed S-ATTack approach.

2.3 Method

In this section, we first explain the notations and definitions. Then, we will provide the details of S-ATTack.

2.3.1 Formulation

Human Trajectory Forecasting

Pedestrian trajectory forecasting addresses a regression task with sequences as inputs and outputs. At any timestep t, the i-th person/agent is represented by his/her xy-coordinates (x_t^i, y_t^i). We denote each agents' observation sequence for T_{obs} timesteps as X^i, a $T_{obs} \times 2$ matrix and the observation sequences of all the n agents in the scene as $X = (X^1, \ldots, X^n)$. Given X, the trajectory predictor f predicts the next T_{pred} positions of all agents $Y = (Y^1, \ldots, Y^n) = f(X^1, \ldots, X^n)$ where Y^i is a $T_{obs} \times 2$ matrix.

Adversarial Examples for Trajectory Forecasting

Equipped with the notations introduced in Section 2.3.1, we will provide a definition of adversarial examples for trajectory forecasting. In this chapter, without loss of generality, we assume the perturbation R is added to the candidate agent which is arbitrarily chosen among agents. Note that in the experiments, all agents are considered as the candidate agent one by one. $\hat{X}^1 = X^1 + R$ while the observations of other agents (which we refer to as neighbors) $\{X^j\}_{j \neq 1}$ remain fixed. Therefore, R is a $T_{obs} \times 2$ matrix of adversarial perturbation, the adversarial example is $\hat{X} = (X^1 + R, X^2, \ldots, X^n)$ and the output of the predictor for that example is $\hat{Y} = (\hat{Y}^1, \ldots, \hat{Y}^n) = f(\hat{X})$. Formally, given a small constant $\epsilon > 0$, a collision distance threshold γ and the maximum of the norm of all rows of a matrix $\|\cdot\|_{\max}$, a socially-attended adversarial example is obtained if:

$$\exists t, j \neq 1, \|R\|_{\max} \leq \epsilon : \left\| \hat{Y}_t^j - \hat{Y}_t^1 \right\| < \gamma. \tag{2.2}$$

In other words, this type of adversarial examples is based on perturbing an observation trajectory so that f predicts the future timesteps with at least a collision (the distance less than γ) between two agents j and 1 in one timestep t. In addition, without loss of generality, we focus on the collisions between the candidate agent and neighbors. Clearly, it can directly be expanded to collisions between any two agents. In the next section, we will describe how we obtain R using Socially-attended attack.

2.3.2 Socially-ATTended ATTack (S-ATTack)

We propose three optimization problems based on different attention mechanisms for socially-attended attack to find suitable perturbations for a collision. The optimization problems are explained in sections 2.3.2, 2.3.2 and 2.3.2 and are solved using an iterative algorithm based on projected gradient descent elaborated in Section 2.3.3.

Before introducing the three optimizations, we will explain the distance matrix which is used in all three optimizations. Given the perturbation R, and a model f, we define the distance matrix $D(R) \in \mathbb{R}^{(n-1)\times T_{pred}}$ as a function of the input perturbation R. It includes the pairwise distance of all neighboring agents from the candidate agent in all prediction timesteps. Let $d_{j,t}$ denote the element at j-th row and t-th column of $D(R)$, i.e., the distance of the agent j from the candidate agent at timestep t of the prediction timesteps:

$$d_{j,t} := \left\| \hat{Y}_t^j - \hat{Y}_t^1 \right\| . \tag{2.3}$$

Hence, for a particular R, the distance matrix $D(R)$ can be leveraged to indicate whether a collision has occurred. We now explain three methods to find such a perturbation by optimizing different cost functions depending on $D(R)$.

No-attention Loss

Our first attempt is to introduce a simple social loss to find perturbations that make collisions in the trajectory forecasting sequences. To achieve that, we find the perturbations that make a collision among the predictions of humans in the prediction model. One naive solution is to minimize the sum of distances between the candidate agent and its neighbors in all prediction timesteps:

$$\min_R \|D(R)\|_F . \tag{2.4}$$

This naive scheme gives the same attention to all agents, which may not be efficient in obtaining small perturbations. For instance, a far agent may not be a good potential candidate for collision and should receive lower attention.

Hard-attention Loss

A better approach to cause a collision is to target a specific agent in a specific timestep instead of taking an average over all neighboring agents. Then, the model is attacked in a way that the distance of the chosen agent with the candidate agent is decreased in the corresponding timestep

until the collision occurs. The equation of the hard-attention attack is as follows:

$$\begin{aligned} &\min_{R,W} \operatorname{Tr}\left(W^\top D(R)\right) + \lambda_r \|R\|_F, \\ &\text{s.t. } w_{j,t} = \delta_{jk}\delta_{tm}, \quad k,m = \operatorname*{argmin}_{j,t} d_{j,t}, \end{aligned} \tag{2.5}$$

where δ is the Kronecker delta function. Besides, W is the attention weight matrix and $w_{j,t}$ is the attention weight for the agent j at timestep t of the prediction timesteps. Indeed, the associated weight to the target agent $w_{k,m}$ which is the closest agent-timestep is 1 and others are 0. The socially-attended loss is the trace (Tr) of multiplication of $D(R)$ by the transpose of W added with the regularization on the perturbation with the balancing coefficient λ_r that encourages finding a small perturbation sequence to make a collision.

Soft-attention Loss

The main drawback of Eq. (2.5) is that the target point (k, m) for a collision is selected based on the assumption that attacking the closest agent-timestep requires small-enough perturbation. This confines the target point selection and might not find the most optimum target point. Note than the models are non-linear and a collision with the closest agent-timestep may not essentially require the smallest perturbation.

To address that, we let the attack attend to the optimal target by itself. We introduce a soft-attention mechanism in which the weights associated to each agent-timestep is assigned by the attack in order to achieve a smaller perturbation. The equation of the soft-attention attack is as follows:

$$\begin{aligned} &\min_{R,W} \operatorname{Tr}\left(W^\top \tanh(D(R))\right) + \lambda_r \|R\|_F - \lambda_w \|W\|_F, \\ &\text{s.t. } \sum_{j,t} w_{j,t} = 1, \quad w_{j,t} \geq 0, \end{aligned} \tag{2.6}$$

where $\tanh$ is applied to the entries of $D(R)$ in order to concentrate less on very far agent-timesteps. Also, we discourage uniformity of weights by subtracting the Frobenius norm of W multiplied by a scalar λ_w.

W is initialized with a uniform distribution. It is progressively updated and puts more weights to the more probable targets for making a collision. Near the convergence point, the best target agent receives a weight value close to 1 while the rest receive 0. Note that W and R are optimized jointly per input sample. We show one example of how W changes in the training in Appendix A.

We will compare our social adversarial attacks (hard-attention Eq. (2.5) and soft-attention Eq. (2.6)) in Section 2.4.3. For brevity, in the rest of the chapter, by S-ATTack, we refer to the attack with soft-attention.

2.3.3 S-ATTack Algorithm

In this section, we explain how adversarial perturbations are achieved using the introduced loss functions. The pseudo-code of the algorithm is written in Algorithm 1. The method is an iterative algorithm with maximum k_{max} iterations. At each iteration, first, perturbed inputs $\hat{X}$ are calculated by adding perturbation R to X^1. Then, we find new predictions $\hat{Y}$ using the perturbed inputs $\hat{X}$. Next, we use projected gradient descent [60] to solve the constraint optimization problems introduced in the previous subsections. Namely, the perturbation R is updated using the gradient of the equations defined above with hyperparameter α and then projected to the ℓ_∞ ball with radius ϵ. Finally, if a collision in the predictions exists, the algorithm stops otherwise it continues until the maximum iterations.

Algorithm 1: The pseudo-code of S-ATTack algorithm

Input: Sequence X, Predictor f
Output: Perturbed sequence $\hat{X}$

Initialize $k \leftarrow 0$, $\hat{X} \leftarrow X$,$R \leftarrow 0$
while $k < k_{max}$ **do**
  $\hat{X} = (X^1 + R, X^2, \ldots, X^n)$
  $\hat{Y} = f(\hat{X})$
  $R = R + \alpha \nabla(Eq.\ (2.4)\ or\ Eq.\ (2.5)\ or\ Eq.\ (2.6))$

  $[R_{i,j}] = [\min(R_{i,j}, \epsilon)]$
  compute D using Eq. (2.3)
  if $\exists\ s \neq 1, t:\ d_{s,t} < \gamma$ **then**
    return $\hat{X}$
  end
  $k = k + 1;$
end

2.4 Experiments

We conduct the experiments to answer the following questions: 1) How vulnerable are the trajectory forecasting models against perturbations on the input sequence? 2) Which of the proposed socially-attended attacks can cause collisions more successfully with smaller perturbations? 3) Are we able to leverage the adversarial examples to improve the robustness of the model? Do they help in better learning the social behavior?

2.4.1 Experimental Setup

Baselines

In order to show the effectiveness of our attack, we conduct our experiments on six well-established trajectory forecasting models.

- **Social-LSTM** [25] (**S-LSTM**): where a social pooling method is employed to model interactions based on shared hidden states of LSTM trajectory encoders.
- **Social-Attention** [30] (**S-Att**): where a self-attention block is in charge of learning interactions between agents.
- **Social-GAN** [29] (**S-GAN**): where a max-pooling function is employed to encode neighbourhood information. They leverage a generative adversarial network (GAN) to learn the distribution of trajectories.
- **Directional-Pooling** [14] (**D-Pool**): where the features of each trajectory is learned using the relative positions as well as the relative velocity and then pool the learned features to learn social interactions.
- **Social-STGCNN** [32] (**S-STGCNN**): where graph convolutional neural network is employed to learn the interactions.
- **PECNet** [61] (**PECNet**): where a self-attention based social pooling layer is leveraged with a variational auto-encoder (VAE) network.

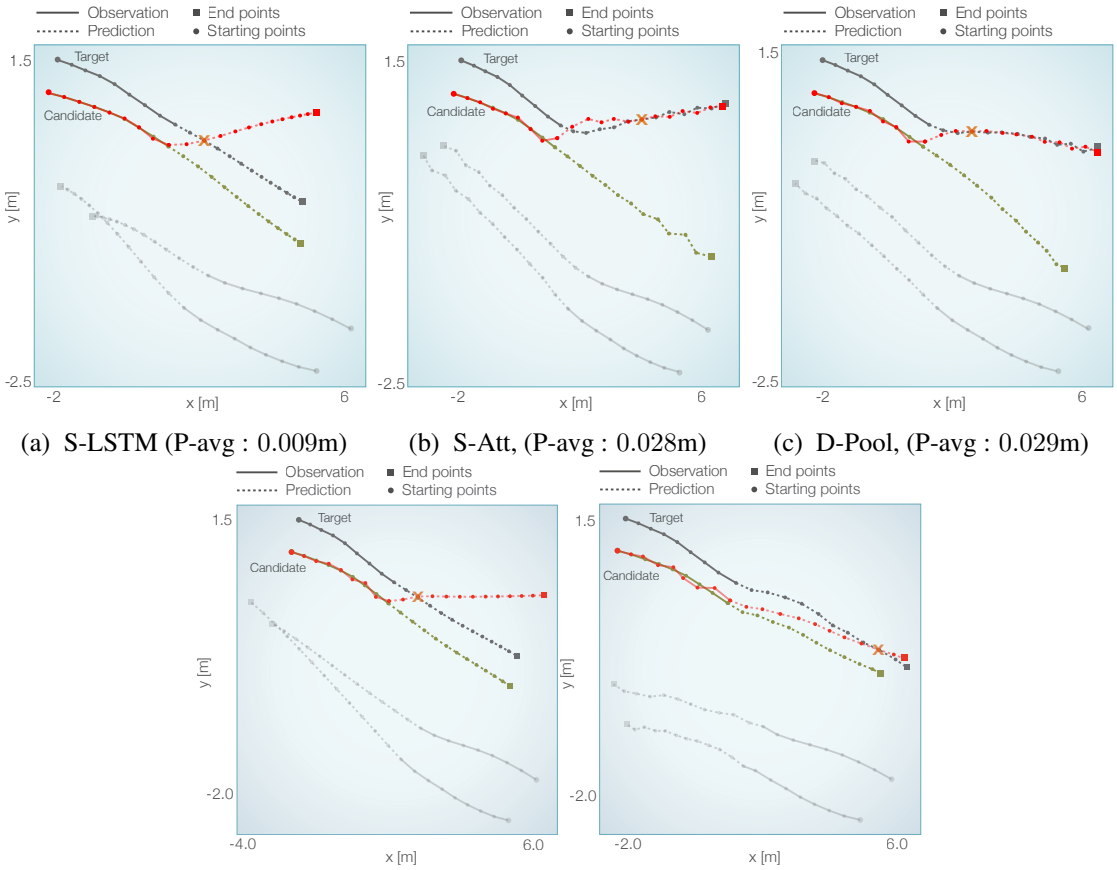
Datasets

ETH [62], UCY [63], and WildTrack [64]: These are well-established datasets with human positions in world-coordinates. We employ two variants of these datasets for our experiments: (1) for S-LSTM, S-Att, S-GAN and D-Pool baselines, we utilized TrajNet++ [14] benchmark which provides identical data splits and data pre-processing. The observation and prediction lengths are considered as 9 and 12, respectively. (2) Since S-STGCNN official code contains its specific pre-processing and data-split on ETH and UCY, we employed the released code to be consistent with their official implementation. Here, the observation and prediction lengths are considered as 8 and 12, respectively.

SDD [65]: The Stanford Drone Dataset is a human trajectory forecasting dataset in bird's eye view. PECNet is one of the state-of-the-art methods with official published code on this dataset. Hence, we report PECNet performance on this dataset. The observation and prediction lengths are considered as 8 and 12, respectively.

Implementation Details

We set the maximum number of iterations to 100. Inspired by the localization algorithm errors mentioned in the introduction, the maximum size of the perturbation for each point ϵ is considered 0.2 m. Also, we set λ_r and λ_w in Eq. (2.5) and Eq. (2.6) equal to 0.1 and 0.5 respectively. The full list of hyperparameters will be provided in Appendix A.



(a) S-LSTM (P-avg : 0.009m) (b) S-Att, (P-avg : 0.028m) (c) D-Pool, (P-avg : 0.029m)

(d) S-GAN, (P-avg : 0.022m) (e) S-STGCNN, (P-avg : 0.042m)

Figure 2.2: Comparison of the performance of different models under our attack. The candidate agent is depicted with green before the perturbation and red after it. For brevity, we have not shown the prediction of neighboring agents before the attack. The scale of y axis is enlarged to better show the difference. The orange X denotes the target point. Our attack achieves collisions with adding small perturbations.

Metrics

In the experiments, we report the performances according to the following metrics:

- Collision Rate (**CR**): this metric measures the existence of collision in the predicted trajectories of the model. Indeed, it calculates the percentage of samples in which at least

Model	Original	Attacked	
	CR [%] ↓	CR [%] ↓	P-avg [m] ↓
S-LSTM [25]	7.8	89.8	0.031
S-Att [30]	9.4	86.4	0.057
S-GAN [29]	13.9	85.0	0.034
D-Pool [14]	7.3	88.0	0.042
S-STGCNN [32]	16.3	59.1	0.11
PECNet [61]	15.0	64.9	0.071

Table 2.1: Comparing the performance of different baselines before (Original) and after the attack (Attacked). Horizontal lines separate models with different datasets.

one collision in the predicted trajectories between the candidate agent and its neighbors occurs. This metric assesses whether the model learned the notion of collision avoidance. Note that we set the distance threshold for indicating a collision γ in Eq. (2.2) equal to 0.2 m .

- Perturbation average (**P-avg**): the average of perturbation sizes at each timestep which is added to the input observation. The numbers are reported in meters.

- Average / Final Displacement Error (**ADE/FDE**): the average/final displacement error between the predictions of the model and the ground-truth values. This metric is commonly used to report the performance of trajectory forecasting models and is reported in meters.

2.4.2 Attack Results

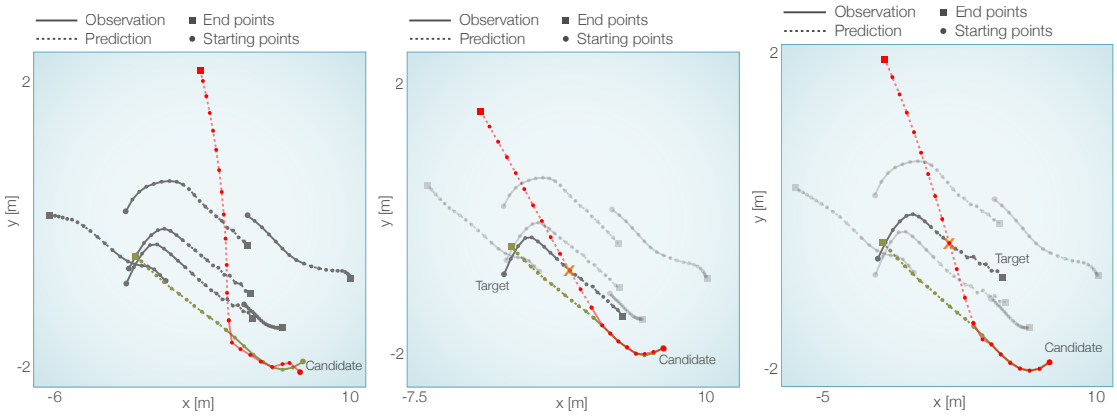
We first provide the quantitative results of applying S-ATTack to the baselines in Table 2.1. The results indicate a substantial increase in the collision rate (at least 3 times) across all baselines by adding perturbations with P-avg smaller than 0.11 m. This questions the social behavior of the models in terms of collision avoidance.

Figure 2.2 visualizes the performance of the baselines S-LSTM, S-Att, D-Pool, S-GAN and S-STGCNN under our attack with the same input. Note that all the baselines are trained on the first group of datasets in 2.4.1. S-LSTM does not change its predictions after adding perturbations but S-Att and D-Pool counteract to avoid collisions. This shows that there exists some collision avoidance behavior understanding in some prediction models, but they are not enough for avoiding a collision. We show more successful cases and also some failure cases in Appendix A.

As D-pool performs better than others in terms of collision avoidance before attack, in the rest of the chapter, we conduct our main experiments on it.

Attacks	CR [%] ↓	P-avg [m] ↓
Random noise	17.6	0.199
No-attention Eq. (2.4)	44.7	0.179
Hard-attention Eq. (2.5)	84.8	0.041
Soft-attention Eq. (2.6)	88.0	0.042

Table 2.2: Comparing different proposed attack methods on D-Pool.



(a) No-attention (P-avg : 0.155m) (b) Hard-attention (P-avg : 0.04m) (c) Soft-attention (P-avg : 0.014m)

Figure 2.3: Comparison of different attack approaches. The no-attention approach Eq. (2.4) could not make a collision despite large P-avg. The hard-attention approach Eq. (2.5) led to a collision with a smaller P-avg. A better target point for collision is found by soft-attention Eq. (2.6) leading to a collision with the least P-avg.

2.4.3 Comparison of Different Attention Methods

In this section, we compare how different strategies for choosing the collision point affects the collision rate and the perturbation size. The quantitative results are shown in Table 2.2. We report the results of Gaussian random noise with variance of 0.2, no-attention Eq. (2.4), hard-attention Eq. (2.5), and soft-attention Eq. (2.6). Random noise and no-attention approaches have small collision rates with large perturbation sizes which indicates the need for a strategy for selecting the collision point. Both hard-attention and soft-attention mechanisms substantially improve collision rate. However, the freedom in the soft-attention approach lets it smartly find better collision points in some of the samples leading to a higher collision rate. Figure 2.3 visualizes the performance of different attack approaches for one data sample. In the illustrated example, in contrast to other two approaches, Soft-attention targets a collision point which is further but easier to collide, thus, leads to a smaller perturbation size.

Target models	Source models			
	S-LSTM	S-Att	S-GAN	D-Pool
S-LSTM [25]	89.8	82.7	85.1	68.3
S-Att [30]	53.0	86.4	57.8	70.0
S-GAN [29]	40.8	59.7	85.0	84.1
D-Pool [14]	88.4	81.3	55.6	88.0
S-Forces [37]	0.69	0.70	1.30	0.60

Table 2.3: Studying the transferability of adversarial examples. The adversarial examples are learned for source models and are transferred to the target models for the evaluation. The reported numbers are Collision Rate (CR) values.

2.4.4 Transferability

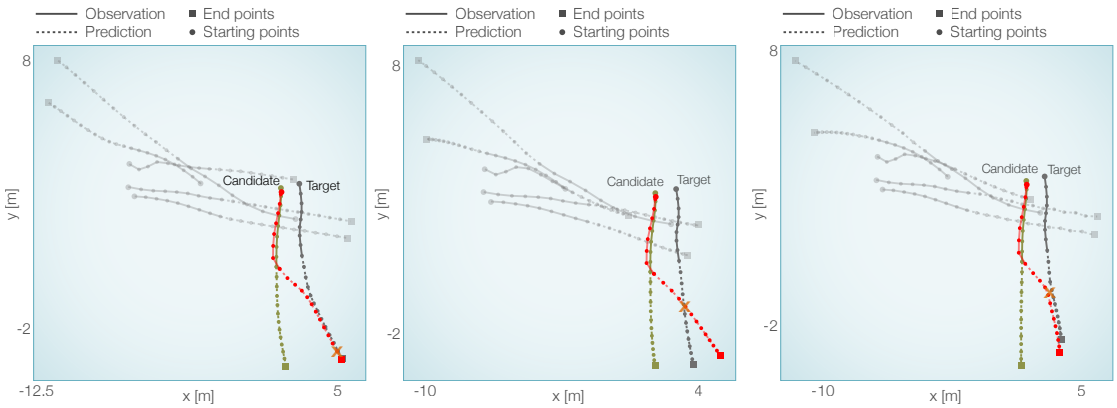
A common practice in adversarial attack studies is to investigate the transferability of perturbations achieved for one model across other models. This shows the existence of common weaknesses across different models or biases in the dataset leading to brittle features in the models. Note that due to the inconsistency of the data used for different baselines, we perform this study only for the models which use the TrajNet++ data.

To study how the generated perturbations can transfer to other models, source models were attacked and the achieved perturbations were used to evaluate others as target models. In addition to the data-driven models, we study the performance of a physics-based model, Social-forces (S-Forces) [37] against generated perturbations. Note that S-Forces has zero collision rate before the attack. Table 2.3 shows that the transferred perturbations can make substantial collision rates for the data-driven models. This means that there exist common defects across different models. Expectedly, S-Forces is robust against the perturbations. This is due to the fact that collision avoidance is an explicit rule defined for the model. However, this robustness comes with the cost of accuracy loss as reported in [25]. We will further analyze our findings in Section 2.4.6. Figure 2.4 shows the result of one identical perturbation on three different models.

2.4.5 Enhancing the Social Understanding

We utilize our S-ATTack to improve the collision avoidance of the model. To this end, we employ a similar approach to [60]. We fine-tune the model using a combination of the original training data and the adversarial examples generated by our S-ATTack method. In this experiment, we set the maximum perturbation size ϵ equal to 0.03.

Table 2.4 indicates that the model's collision avoidance could improve by 11%. Moreover, the collision rate after attack improves by 60% meaning that it is much less vulnerable to the attack. As shown in the table, fine-tuning the model with random noise could not improve the collision avoidance. Therefore, we conclude that our adversarial examples provide useful information to



(a) D-Pool (b) S-LSTM (c) S-Att

Figure 2.4: Transferring an adversarial example obtained from attacking D-Pool model to S-LSTM and S-Att models. P-avg is 0.151 m. We can observe that the perturbation generates collisions in both S-LSTM and S-Att models, although not optimized for them.

		Original		Attacked	
	ADE/FDE [m] ↓	CR [%] ↓	CR gain [%] ↑	CR [%] ↓	CR gain [%] ↑
D-Pool	0.57 / 1.23	7.3	-	37.3	-
D-Pool w/ rand noise	0.57 / 1.23	7.5	-2.7	36.1	+3.2
D-Pool w/ S-ATTack	0.60 / 1.28	**6.5**	**+10.4**	**14.7**	**+60**

Table 2.4: Comparing the original model and the fine-tuned model with random-noise data augmentation (D-Pool w/ rand noise) and S-ATTack adversarial examples (D-Pool w/ S-ATTack). The numbers are on TrajNet++ challenge testset. ADE, FDE are reported in meters.

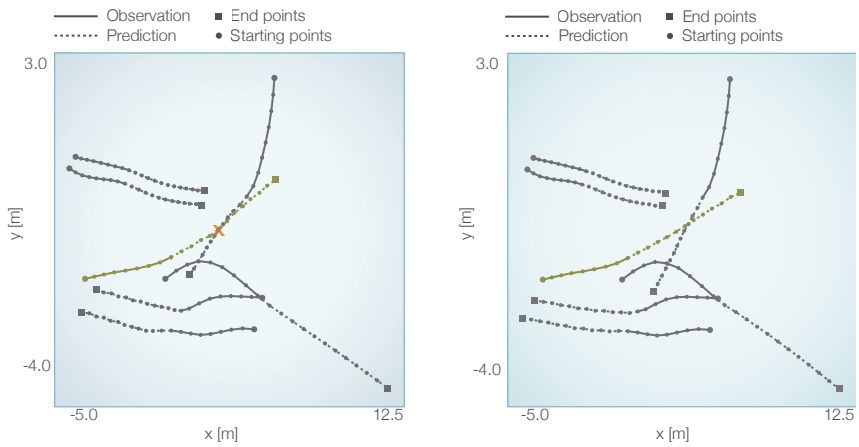
improve the collision avoidance of the model. Note that the prediction error of the model in terms of ADE/FDE is slightly increased. This shows a trade-off between accuracy and robustness. This can be similar to the findings in the previous works on image classifiers [66].

We also show the performance of these models visually. In Figure 2.5 (a), the original model's prediction with an occurring collision is seen while the enhanced model prediction is collision-free (Figure 2.5 (b)). In the next figure, Figure 2.6 (a), we can observe that attacking D-Pool leads to a collision with a P-avg of 6.6 but the enhanced model is robust against attack (Figure 2.6 (b)) indicating a better collision avoidance understanding.

2.4.6 Discussions

In this section, equipped with the performed experiments in previous parts, we want to shed light on the weaknesses of the studied trajectory forecasting models. Also, we will mention the limitations of our work.

1. Although the trajectory forecasting models are designed to capture interactions among



(a) D-Pool Original

(b) D-Pool w/ S-ATTack Original

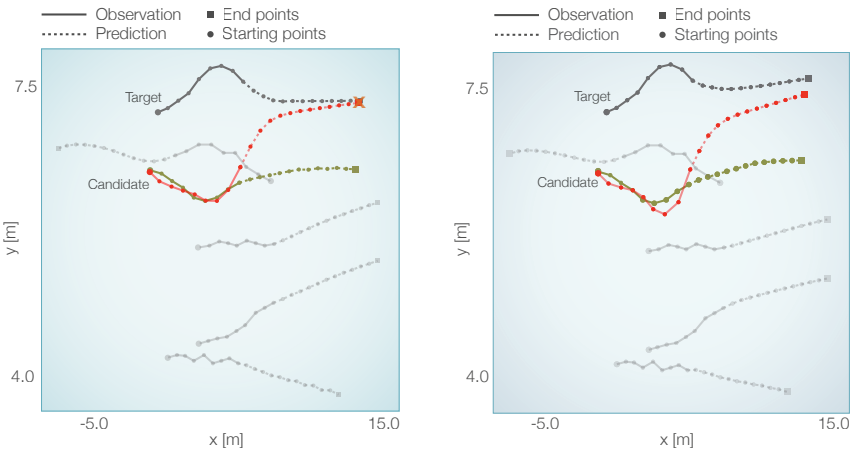
Figure 2.5: Comparison of the performance of the original model and the enhanced one with S-ATTack. The original model has a collision in its predictions (a) while the enhanced model prediction is collision-free (b).

people, our results in Section 2.4.2 showed that they are fragile and cannot generalize to perturbed data. Moreover, we showed in Section 2.4.4 that the perturbations are transferable across different models indicating the existence of common weaknesses across these models. We have two hypotheses as the reasons of high vulnerability to the attack and transferability of the perturbations: 1) this can be due to the lack of collision avoidance inductive bias in the models. While inductive bias is more influential when the size of the training data is limited, it may be of less impact in large data regime. 2) because of limited training data, there exists unexplored input space for the predictor model. We leave a detailed study on these points for future work.

2. The success of our attack in making collisions may raise the concern that the models are actually unaware of the social interactions. We perform an experiment to assess this point. We attack the model while keeping the predictions of neighboring agents frozen for the attack algorithm. Therefore, the attack cannot consider other agents' counteracts to the perturbations. As shown in Figure 2.7, the attack fails in causing collision. This result indicates that while S-ATTack can cause collisions in the models' predictions, the models have limited social understanding allowing them to counteract perturbations.

3. To compare the models' sensitivity to the perturbations added to different observation points, we add small random noise (less than 0.2m) to each point separately and measure its effect on the prediction error. Figure 2.8 shows that the predictors are over-dependant on the last observation point. It is consistent with our findings that the perturbations generated by S-ATTack tend to have larger components on the last timesteps trying to use this feature.

4. Are different prediction timesteps equally vulnerable to the collisions? To answer this question, we consider a scenario where two pedestrians are crossing each other at different



(a) D-Pool Attacked

(b) D-Pool w/ S-ATTack Attacked

Figure 2.6: Comparison of the vulnerability of the victim model under attack and the enhanced one with S-ATTack. While D-Pool can be attacked to cause a collision with a P-avg of 6.6 (a), the enhanced model cannot be attacked even with a large perturbation (b).

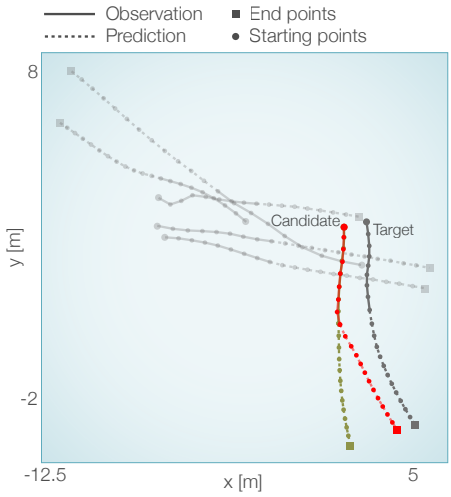


Figure 2.7: We froze the predictions of neighboring agents for the attack algorithm. The algorithm fails in making collision since it is unaware of the counteracts of neighbors to the perturbations. This indicates that the prediction model counteracts with the perturbations. Also, the attack algorithm requires to consider other agents' counteracts to cause a collision. The same sample was successfully attacked in Figure 2.4.

prediction timesteps and measure the collision rate. We use S-Forces,S-LSTM and D-Pool. Figure 2.9(a) visualizes the predictions of S-LSTM and S-Forces models for one scenario. While S-LSTM is unable to avoid the collision, S-Forces decreases the prediction speed in the first points to avoid the collision. Then, we study the sensitivity of models in different timesteps with regards to collision avoidance. To this end, we run scenarios like

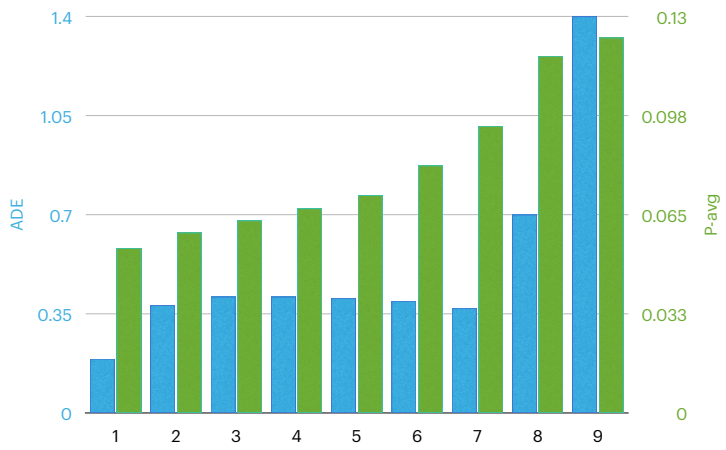
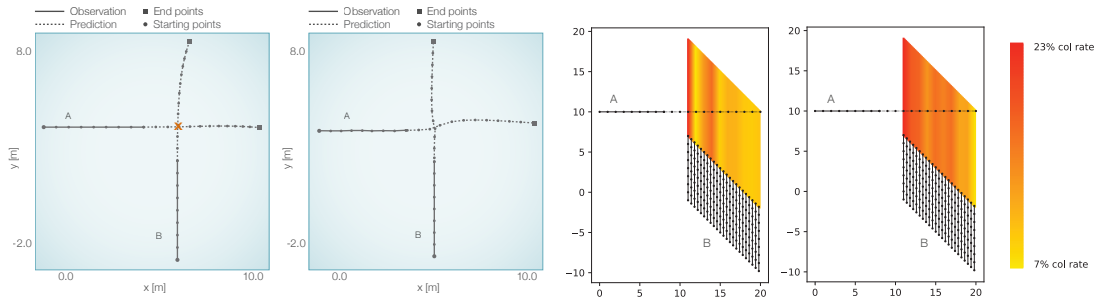


Figure 2.8: Two overlaid figures analyzing the impact of different timesteps. The blue is the average change in the predictions by adding a random noise of size 0.2m to different observation timesteps. The green curve shows the perturbation size at each timestep in perturbations found by S-ATTack.

Figure 2.9(a),(b) 1000 times for each timestep where small Gaussian noise is added to the observations. The result are shown in a heatmap in Figure 2.9(c). First, a high collision rate is observed for both models revealing their weaknesses in capturing agent-agent interactions. Second, each model has a different pattern of sensitivity with respect to the timestep. Initial timesteps have higher collision rate since it is more difficult to change the manoeuvre in the first timesteps. While smaller collision rates are expected for later timesteps, in both models, the collision rate does not monotonically decrease. This shows that the models are more vulnerable in the middle timesteps. This can be due to the biases in the data or model structure.

5. Our attack perturbs only the candidate agent to achieve a collision. The same method can be employed to perturb other agents. Figure 2.10 shows three different scenarios in a scene where a collision occurs by perturbing different agents. However, collision types of Figure 2.10(b) and Figure 2.10(c) are not considered in our method and can be addressed as a future work.

6. The perturbations are calculated through a joint optimization problem and there is no constraint on the smoothness of the perturbations. To study the impact of our adversarial attack in the presence of a smoothing filter, a 4th order polynomial is fit to smooth the trajectories according to [67]. In the first and second rows of Table 2.5, the performance of the original model and with S-attack is observed. Next, we add the smoothing in defense (3rd row) and in both defense and attack (4th row). Smoothing in defense makes the final calculated perturbations smooth before feeding to the predictor. Smoothing the trajectory reduces collision but still the attack is highly effective. In the third experiment, we include the smoothing in the attack optimization algorithm where the perturbation is smoothed after



(a) S-LSTM (b) S-Forces (c) Prediction Heatmap

Figure 2.9: Collision rate analysis for different timesteps. We cross two pedestrians A,B to see if the prediction models can avoid the facing collision. (a,b) We visualize the performance of S-LSTM and S-Forces models in a scenario where two agents cross in the third prediction timestep. S-LSTM is not avoiding the collision whereas S-Forces avoids it due to the hand-crafted knowledge of collision avoidance. (c) Analysis of collision rates of two data-driven model, D-Pool (left) and S-LSTM (right) on different timesteps. We cross two pedestrians A,B on different timesteps and report the collision rate for each of them. Dark yellow is the observation sequence for both agents and green is the prediction of A. The red and yellow colors show high and low collision rates, respectively. We observe different patterns for each model. We do not visualize the first two prediction timesteps as the model is unable to change the manoeuvre instantly.

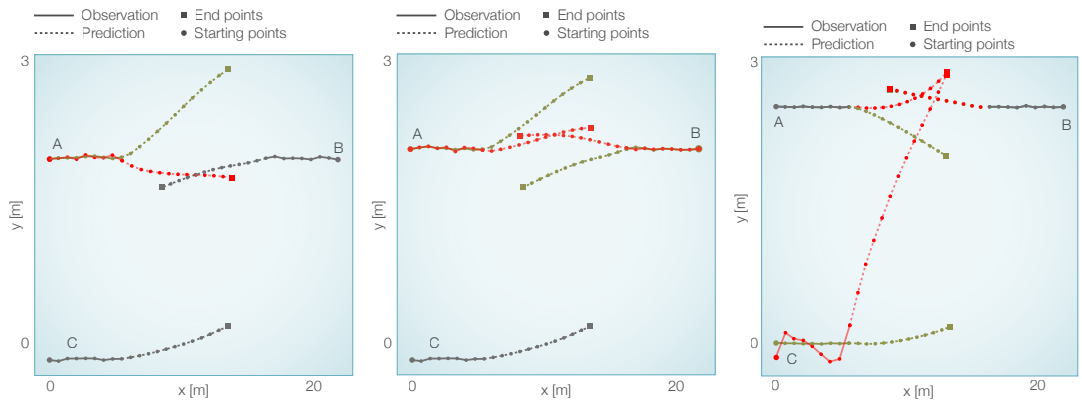
Attacks	CR [%] ↓
D-pool	7.3
D-pool + S-attack	88.0
D-pool + S-attack + Smoothing in defense	61.6
D-pool + S-attack with smoothing + Smoothing in defense	72.0

Table 2.5: Leveraging smoothing function in the attack. Smoothing in defense makes the perturbed observation smooth before feeding to the predictor which can be considered as a defense method against the attack. S-attack with smoothing includes the smoothing function in the optimization problem to achieve a smoothed perturbation.

each optimization iteration. Therefore, S-attack finds a smooth perturbation which shows higher collision rate compared to the previous experiment. Note that CR is lower than the original experiment which is because of the additional polynomial constraint enforced that reduces the search space.

2.5 Conclusions

In this chapter, we studied the robustness properties of trajectory forecasting models in terms of social understanding under adversarial attack. We introduced our Socially-ATTended ATTack (S-ATTack) to cause collisions in state-of-the-art prediction models with small perturbations.



(a) Agent A perturbed (b) Agent A and B perturbed (c) Agent C perturbed

Figure 2.10: Perturbing different agents in a scenario to achieve collision between agents A and B. (a) Perturbing A which is the candidate agent. (b) Perturbing both A and the other agent B that is colliding with. (c) Perturbing C as a third agent who is not colliding with A but affects the predictions of A and B. In all cases, a collision between A and B is predicted. This shows that S-attack is effective for other problem formulations.

Adversarial training using S-ATTack can not only make the models more robust against adversarial attacks, but also reduce the collision rate and hence, improve their social understanding. This chapter sheds light on the common weaknesses of trajectory forecasting models opening a window toward their social understandings.

Our work is a first step that highlights the lack of social understanding in the models. The attack method can be extended by considering the velocity that a collision occurs with, or considering other notions of social behavior such as grouping. To approach socially-aware predictors, we believe that the field lacks two main components. First, using more socially-related metrics instead of ADE/FDE for training and evaluation. This is an area of research where metrics like the time-to-collision metric [68], are required to be studied more. Second, supervised learning in the absence of proper inductive biases cannot learn social behaviors properly, especially with the current small-scale datasets. One direction to address the challenge is to combine physics-based models and neural networks. Also, having more challenging and large-scale datasets can help the models learn the bias by themselves.

In the next chapter, we shift our attention from evaluating social context to evaluating scene context in trajectory forecasting. This is a critical pivot as it allows us to understand and improve how models perceive and react to their physical environment, especially under adversarial conditions.

3 Scene-aware Trajectory Forecasting

This chapter is based on the article:

Mohammadhossein Bahari*, Saeed Saadatnejad*, Ahmad Rahimi, Mohammad Shaverdikondori, Seyed-Mohsen Moosavi-Dezfooli and Alexandre Alahi, *Vehicle trajectory prediction works, but not everywhere*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022

The code and a summary video of the work can be found on the project's webpage[1].

3.1 Introduction

Picking up from where we left off in the previous chapter, we continue our evaluation of trajectory forecasting models by shifting our focus to scene-awareness. Consideration of scene awareness becomes even more pertinent in the context of vehicle motion, although it retains its relevance for human motion too.

Vehicle trajectory forecasting is one of the main building blocks of a self-driving car, which forecasts how the future might unfold based on the road structure (*i.e.,* the scene) and the traffic participants. State-of-the-art models are commonly trained and evaluated on datasets collected from a few cities [69]–[71]. While their evaluation has shown impressive performance, *i.e.*, almost no off-road prediction, their generalization to other types of possible scenes *e.g.*, other cities, remains unknown. Figure 3.1 shows a real-world example where a state-of-the-art model reaching zero off-road in the known benchmark [70] failed in *South St, New York, USA*. Since collecting and annotating data of all real-world scenes is not a viable and affordable solution, we present a method that automatically investigates the robustness of vehicle trajectory forecasting to the scene. We tackle the problem through the lens of realistic adversarial scene generation.

Given an observed scene, we want to generate a realistic modification of it such that the prediction

[1] https://s-attack.github.io/

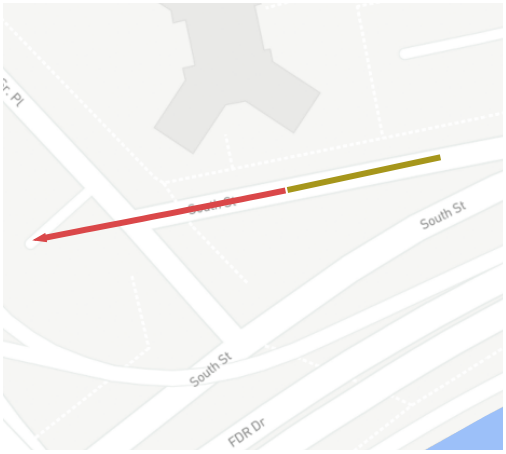


Figure 3.1: A real-world place (location) in New York where the trajectory forecasting model (here [15]) fails. We find this place by retrieving real-world locations which resemble our conditional generated scenes for the prediction model.

models fail in. Having an off-road prediction is a clear indication of a failure in the the model's scene reasoning and has been used in some previous works [72]–[75]. To find a realistic example where the models go off-road, the huge space of possible scenes should be explored. One solution is data-driven generative models that mimic the distribution of a dataset [76]. Yet, they do not essentially produce realistic scenes due to the possible artifacts. Moreover, they will represent a portion of real-world scenes as they cannot generate scenes beyond what they have observed in the dataset (cannot extrapolate). We therefore suggest a simple yet efficient alternative. We show that it is possible to use a limited number of simple functions for transforming the scene into new realistic but challenging ones. Our method can explicitly extrapolate to new scenes.

We introduce atomic scene generation functions where given a scene in the dataset, the functions generate multiple new ones. These functions are chosen such that they can cover a range of realistic scenes. We then choose the scenes where the prediction model produces an off-road trajectory. Using three state-of-the-art trajectory forecasting models trained on Argoverse public dataset [70], we demonstrate that more than 60% of the existing scenes in the dataset can be modified in such a way that it will make state-of-the-art methods fail (*i.e.,* predict off-road). We confirm that the generated scenes are realistic by finding real-world locations that partially resemble the generated scenes. We also demonstrate off-road predictions of the models in those locations. To this end, we extract appropriate features from each scene and use image retrieval techniques to search public maps [77]. We finally show that these generated scenes can be used to improve the robustness of the models.

Our contributions are fourfold:

- we highlight the need for a more in-depth evaluation of the robustness of vehicle trajectory forecasting models;

- our work proposes an open-source evaluation framework through the lens of realistic adversarial scene generation by promoting an effective generative model based on atomic scene generation functions;
- we demonstrate that our generated scenes are realistic by finding similar real-world locations where the models fail;
- we show that we can leverage our generated scenes to make the models more robust.

3.2 Related Work

3.2.1 Vehicle Trajectory Forecasting

The scene plays an important role in vehicle trajectory forecasting as it constrains the future positions of the agents. Therefore, modeling the scene is common compared to some human trajectory forecasting models. In order to reason over the scene in the predictions, some suggested using a semantic segmented map to build circular distributions and outputting the most probable regions [78]. Another solution is reasoning over raw scene images using convolutional neural networks (CNN) [79]. Many follow-up works represented scenes in the segmented image format and used the learning capability of CNNs over images to account for the scene [80]–[84]. Carnet [85] used attention mechanism to determine the scene regions that were attended more, leading to an interpretable solution. Some recent work showed that scene can be represented by vector format instead of images [15], [86]–[88]. To further improve the reasoning of the model and generate predictions admissible with respect to the scene, use of symmetric cross-entropy loss [72], [89], off-road loss [74], and REINFORCE loss [73] have been proposed. Despite all these efforts, there has been limited attention to assess the performance of trajectory forecasting models on new scenes. Our work proposes a framework for such assessments.

3.2.2 Evaluating Autonomous Driving Systems in Scene Context

Self-driving cars deal with dynamic agents nearby and the static environment around. Several works studied the robustness of self-driving car modules with respect to the status of dynamic agents on the road, *e.g.*, other agents. Some previous works change the behavior of other agents in the road to act as attackers and evaluate the model's performance with regards to the interaction with other agents as we described in the previous chapter and other works [90]–[95]. Others directly modify the raw sensory inputs to change the status of the agents in an adversarial way [96]–[99].

In addition to the dynamic agents, driving is highly dependent on the static scene around the vehicle. The scene understanding of the models can be assessed by modifying the input scene. Previous works modify the raw sensory input by changing weather conditions [100]–[102], generating adversarial drive-by billboards [103], [104], and adding carefully crafted patches/lines

to the road [105], [106]. These works have not changed the shape of the scene, *i.e.*, the structure of the road. In contrast, we propose a conditional scene generation method to assess the scene reasoning capability of trajectory forecasting models. Also our approach is different from data-driven scene generation based on graph [76] or semantic maps [21]. Data-driven generative models are prone to have artifacts and cannot extrapolate beyond the training data. Ours is an adversarial one which can extrapolate to new scenes.

3.3 Method

In this section, we explain in detail our approach for generating realistic scenes. After introducing the notations in Section 3.3.1, we show how we generate each scene in Section 3.3.2 and satisfy physical constraints in Section 3.3.3. Finally, we introduce our search method in Section 3.3.4.

3.3.1 Problem Setup

The vehicle trajectory forecasting task is usually defined as predicting the future trajectory of a vehicle z given its observation trajectory h, status of surrounding vehicles a, and scene S. For the sake of brevity, we assume S is in the vector representation format [70][1] [2]. Specifically, S is a matrix of stacked $2d$ coordinates of all lanes' points in x-y coordinate space where each row represents a point $s = (s_x, s_y)$. Formally, the output trajectory z of the predictor g is:

$$z = g(h, S, a). \tag{3.1}$$

Given a scene S, our goal is to create challenging realistic scene S^* as we will explain in Section 3.3.2.

3.3.2 Conditional Scene Generation

Our controllable scene generation method generates diverse scenes conditioned on existing scenes. Specifically, we opt for a set of atomic functions which represent turn as a typical road topology. To this end, we normalize the scene (*i.e.*, translation and rotation with respect to h), apply the transformation functions, and finally denormalize to return the generated scene to the original view. Note that every transformation of S is followed by the same transformations on h and a.

We define transformations on each scene point in the following form:

$$\tilde{s} = (s_x, s_y + f(s_x - b)) \tag{3.2}$$

where $\tilde{s}$ is the transformed point, f is a single-variable transformation function, and b is the border parameter that determines the region of applying the transformation. In other words, we

[21] We show in Appendix B.4 that our method is seamlessly applicable when S is in image representation.

define $f(<0)=0$ so the areas where $s_x < b$ are not modified. This confines the changes to the regions containing the prediction. One example is shown in Figure 3.2. The new scene is named $\tilde{S}$, a matrix of stacked $\tilde{s}$. We propose three interpretable analytical functions for the choice of f.

Smooth-turn: this function represents different types of single turns in the road.

$$\begin{aligned} f_{st,\alpha}(s_x) &= \begin{cases} 0, & s_x < 0 \\ q_\alpha(s_x), & 0 \leq s_x \leq \alpha_1 \\ (s_x - \alpha_1)q'_\alpha(\alpha_1) + q_\alpha(\alpha_1) & \alpha_1 < s_x \end{cases}, \\ q_\alpha(s_x) &= \alpha_2 s_x^{\alpha_3}, \\ \alpha &= (\alpha_1, \alpha_2, \alpha_3), \end{aligned} \tag{3.3}$$

where α_1 determines the length of the turn, α_2, α_3 control its sharpness, and q'_α indicates the derivative of the defined auxiliary function q_α. Note that according to the definition, $f_{st,\alpha}$ is continuously differentiable and makes a smooth turn. One such turn is depicted in Figure 3.2b.

Double-turn: these functions represent two consecutive turns with opposite directions. Also, there is a variable indicating the distance between them:

$$\begin{aligned} f_{dt,\beta}(s_x) &= f_{st,\beta_1}(s_x) - f_{st,\beta_1}(s_x - \beta_2), \\ \beta &= (\beta_{11}, \beta_{12}, \beta_{13}, \beta_2), \\ \beta_1 &= (\beta_{11}, \beta_{12}, \beta_{13}), \end{aligned} \tag{3.4}$$

where β_1 is the set of parameters of each turn described in Equation (3.3) and β_2 is the distance between two turns. One example is shown in Figure 3.2c.

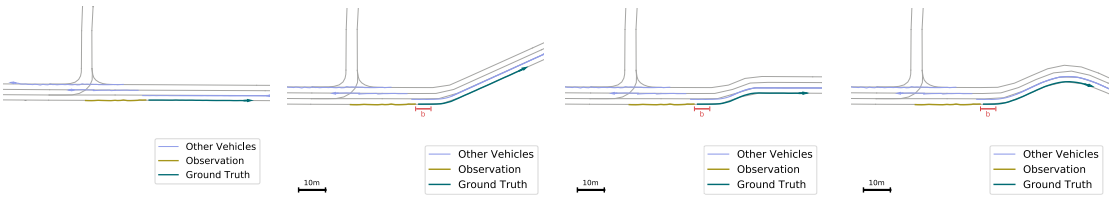
Ripple-road: one type of scene that can be challenging for the prediction model is ripple road:

$$\begin{aligned} f_{rr,\gamma}(s_x) &= \begin{cases} 0, & s_x < 0 \\ \gamma_1(1 - cos(2\pi\gamma_2\, s_x)), & s_x \geq 0 \end{cases}, \\ \gamma &= (\gamma_1, \gamma_2), \end{aligned} \tag{3.5}$$

where γ_1 determines the turn curvatures and γ_2 determines the sharpness of the turns. One such turn is depicted in Figure 3.2d.

3.3.3 Physical Constraints

Every scenario consists of a scene and vehicle trajectories in it. The generated scenarios must be feasible, otherwise, they cannot represent possible real-world cases. We consider a scenario as feasible if a human driver can pass it safely. This means that the physical constraints – *i.e.*, the



(a) Before transformation (b) Single-turn (c) Double-turn (d) Ripple-road

Figure 3.2: Visualization of different transformation functions. The scene before transformation will be followed by three different transformations. Here, $\alpha = (10, 0.002, 3)$ for the single-turn, $\beta = (10, 0.002, 3, 10)$ for the double-turn and $\gamma = (6, 0.017)$ for the ripple-road. b is the border parameter and set to 5 meters in all figures.

Newton's law – should not be violated. The Newton's law indicates a maximum feasible speed for each road based on its curvature [107]:

$$v_{max} = \sqrt{\mu g R}, \tag{3.6}$$

where R is the radius of the road, μ is the friction coefficient and g is the gravity. To consider the most conservative situation, we pick the maximum curvature (minimum radius) existing in the generated road. Then, we slow down the history trajectory when the speed is higher than the maximum feasible speed, and we name it $\tilde{h}$. Note that this conservative speed scaling ensures a feasible acceleration too. We will show in Section 3.4 that a model with hard-coded physical constraints successfully predicts the future trajectory for the generated scenes, which indicates that our constraints are enough.

3.3.4 Scene Search Method

In the previous sections, we defined a realistic controllable scene generation method. Now, we introduce a search method to find a challenging scene specific to each trajectory forecasting model.

We define m as a function of z and S measuring the percentage of prediction points that are off-road obtained using a binary mask of the drivable area. We aim to solve the following problem to find a scene in which the prediction model fails in:

$$\begin{aligned} S^* &= \arg\min_{\tilde{S}} l(\tilde{z}, \tilde{S}), \\ l(\tilde{z}, \tilde{S}) &= \Big(1 - m(\tilde{z}, \tilde{S})\Big)^2, \end{aligned} \tag{3.7}$$

where $\tilde{S}$ is a modification of S according to Equation (3.2) using one of the transformation functions Equation (3.3), Equation (3.4), or Equation (3.5). Moreover, $\tilde{z} = g(\tilde{h}, \tilde{S}, \tilde{a})$ is the model's predicted trajectory given the modified scene and the modified history trajectories.The optimization problem finds the corresponding parameters to obtain S^* that gives the highest number

of off-road prediction points. Equation (3.7) can be optimized using any black-box optimization technique. We have studied Bayesian optimization [108], [109], Genetic algorithms [110], [111], Tree-structured Parzen Estimator Approach (TPE) [112] and brute-force. The overall algorithm is described in Appendix B.3.

3.4 Experiments

We conduct experiments to answer the following questions: 1) How is the performance of the prediction models on our generated scenes? 2) Are the generated scenes realistic and possibly similar to the real-world scenes? 3) Are we able to leverage the generated scenes to improve the robustness of the model?

3.4.1 Experimental Setup

Baselines and Datasets

We conduct our experiments on the baselines with different scene reasoning approaches (lane-graph attention [15], symmetric cross entropy [72], and counterfactual reasoning [113]), which are among the top-performing models and are open-source.

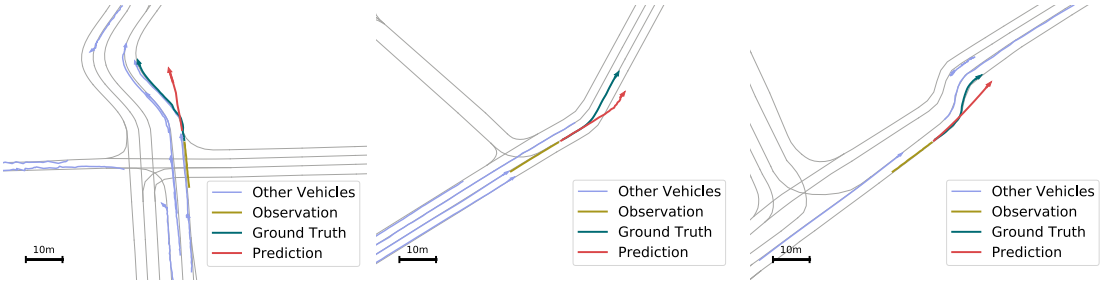
LaneGCN [15]. It constructs a lane graph from vectorized scene and uses self-attention to learn the predictions. This method was among the top methods in Argoverse Forecasting Challenge 2020 [114]. It is a multi-modal prediction model which also provides the probability of each mode. Therefore, in our experiments, we consider the mode with the highest probability.

DATF [72]. It is a flow-based method which uses a symmetric cross-entropy loss to encourage producing on-road predictions. This multi-modal prediction model does not provide the probability of each mode. We therefore consider the mode which is closest to the ground truth.

WIMP [113]. They employ a scene attention module and a dynamic interaction graph to capture geometric and social relationships. Since they do not provide probabilities for each mode of their multi-modal predictions, we consider the one which is closest to the ground truth.

MPC [22], [115]. We report the performance of a rule-based model with satisfied kinematic constraints. We used a well-known rule-based model which follows center of the lanes [115]. While many approaches can be used to satisfy kinematic constraints in trajectory forecasting, similar to [22], we used Model Predictive Control (MPC) with a bicycle dynamic model.

We leveraged Argoverse dataset [70], the same dataset our baselines were trained on. Given the 2 seconds observation trajectory, the goal is to predict the next 3 seconds as the future motion of the vehicle. It is a large scale vehicle trajectory dataset. The dataset covers parts of Pittsburgh and Miami with total size of 290 kilometers of lanes.



(a) DATF (b) WIMP (c) LaneGCN

Figure 3.3: The predictions of different models in some generated scenes. All models are challenged by the generated scenes and failed in predicting in the drivable area.

Metrics

Hard Off-road Rate (**HOR**): in order to measure the percentage of samples with an inadmissible prediction with regards to the scene, we define HOR as the percentage of scenarios that at least one off-road happens in the prediction trajectory points. It is rounded to the nearest integer.

Soft Off-road Rate (**SOR**): to measure the performance in each scenario more thoroughly, we measure the percentage of off-road prediction points over all prediction points and the average over all scenarios is reported. The reported values are rounded to the nearest integer.

Implementation Details

We set the number of iterations to 60, the friction coefficient μ to 0.7 [116] and b equal to 5 for all experiments. For the choice of the black-box algorithm, as the search space of parameters is small in our case, we opt for the brute-force algorithm. We developed our model using a 32GB V100 NVIDIA GPU.

3.4.2 Results

We first provide the quantitative results of applying our method to the baselines in Table 3.1. The last column (All) represents the results of the search method described in Section 3.3.3. We also reported the performance of considering only one category of scene generation functions in the optimization problem Equation (3.7) in the other columns of the table. The results indicate a substantial increase in SOR and HOR across all baselines in different categories of the generated scenes. This shows that the generated scenes are difficult for the models to handle. LaneGCN and WIMP have competitive performances, but WIMP run-time is 50 times slower than LaneGCN. Hence, we use LaneGCN to conduct our remaining experiments.

Figure 3.3 visualizes the performance of the baselines in our generated scenes. We observe that all models are challenged with the generated scenes. More cases are provided in Appendix B.1.

Model	Original	Generated (**Ours**)			
		Smooth-turn	Double-turn	Ripple-road	All
	SOR / HOR	SOR / HOR	SOR / HOR	SOR / HOR	SOR / HOR
DATF [72]	1 / 2	37 / 77	36 / 76	42 / 80	43 / 82
WIMP [113]	0 / 1	13 / 46	14 / 50	20 / 58	22 / 63
LaneGCN [15]	0 / 1	8 / 40	19 / 60	21 / 62	23 / 66
MPC [115]	0 / 0	0 / 0	0 / 0	0 / 0	0 / 0

Table 3.1: Comparing the performance of different baselines in the original dataset scenes and our generated scenes. SOR and HOR are reported in percent and the lower represent a better reasoning on the scenes by the model. MPC as a rule-based model always has on-road predictions both in original and our generated scenes.

Model	w/ phys	w/o phys
	SOR / HOR	SOR / HOR
LaneGCN	33 / 85	47 / 92
MPC	0 / 1	0 / 3

Table 3.2: Impact of the physical constraints. We report the performance with and without the physical constraints explained in Section 3.3.3. The numbers are reported on samples of data with speed higher than v_{max} in their h.

In Table 3.1, we observe that SOR is less than or equal to 1% for all methods in the original scenes. Our exploration shows that more than 90% of these off-road cases are due to the annotation noise in the drivable area maps of the dataset and the models are almost error-free with respect to the scene. Some figures are provided in Appendix B.1. While this might lead to the conclusion that the models are flawless, results on the generated scenes question this conclusion. We confirm our claim in the next section by retrieving the real-world scenes where the model fails.

Feasibility of a scenario is an important feature for generated scenes. As mentioned in Section 3.3.3, we added physical constraints to guarantee the physical feasibility of the scenes. Table 3.1 indicates that MPC as a rule-based model predicts almost without any off-road in the generated scenarios. It approves that the scenes are feasible with the given history trajectory. In order to study the importance of added constraints, we relax the constraints for the generated scenes. We report the performance of the baseline and MPC on the cases where the maximum speed in their h is higher than v_{max}. In Table 3.2, we observe that without those feasibility-assurance constraints, there are more cases where MPC is unable to follow the road and has $3\times$ more off-road. We conclude that those constraints are necessary to make the scene feasible. We keep the constraints in all of our experiments to generate feasible scenarios.

(a) Paris location (b) Hong Kong location (c) New Mexico location

Figure 3.4: Retrieving some real-world locations similar to the generated scenes using our real-world retrieval algorithm. We observe that the model fails in Paris (a), Hong Kong (b) and New Mexico (c).

3.4.3 Real-world Retrieval

So far, we have shown that the generated scenes along with the constraints are feasible/realistic scenes. Next, we want to study the plausibility/existence of the generated scenes. Inspired by image retrieval methods [117], we develop a retrieval method to find similar roads in the real-world. First, we extract data of 4 arbitrary cities (New York, Paris, Hong Kong, and New Mexico) using OSM [77]. Then, $20,000$ random samples of 200×200 meters are collected from each city. Note that it is the same view size as in Argoverse samples. Then, a feature extractor is required to obtain a feature vector for each scene. We used the scene feature extractor of LaneGCN named MapNet to obtain some 128 dimensional feature vectors for each sample. We then use the well-known image retrieval method K-tree algorithm [117]. It first uses K-Means algorithm multiple times to cluster the feature vectors of all scenes into a predefined number of clusters (in our case 1000). Then, given a generated scene as the query, it sorts real scenes based on the similarity with the query scene and retrieves 10 closest scenes to the query. Finally, we test the prediction model in these examples. Some examples are provided in Figure 3.4. More scenes can be found Appendix B.1.

3.4.4 Robustness

Here, we study if we can make the models robust against new generated scenes. To this end, we fine-tune the trained model using a combination of the original training data and the generated examples by our method for 10 epochs.

We report the performance of these models in the generated scenes with different transformation power. Transformation power is determined by $\alpha_2 \times 3000$, $\beta_{12} \times 3000$ and γ_1 for Equation (3.3), Equation (3.4), and Equation (3.5), respectively. It represents the amount of curvature in the scene. Table 3.3 indicates that without losing the performance in the original accuracy metrics, the

Model	Pow=1 SOR/HOR	Pow=3 SOR/HOR	Pow=5 SOR/HOR	Pow=7 SOR/HOR	Pow=9 (Full) SOR/HOR
LaneGCN	2 / 8	12 / 35	19 / 49	22 / 58	23 / 66
LaneGCN w/ aug	**1 / 7**	**6 / 21**	**10 / 30**	**13 / 38**	**14 / 46**

Table 3.3: Comparing the original model and the fine-tuned model with data augmentation of the generated scenes. The performance is reported on generated scenes with different transformation power (Pow). Transformation power is determined by $\alpha_2 \times 3,000$, $\beta_{12} \times 3,000$ and γ_1 for Equation (3.3), Equation (3.4), and Equation (3.5), respectively which represents the amount of curvature in the scene. The average / final displacement errors on original scenes are equal to $1.35/2.98m$ for both original and fine-tuned models.

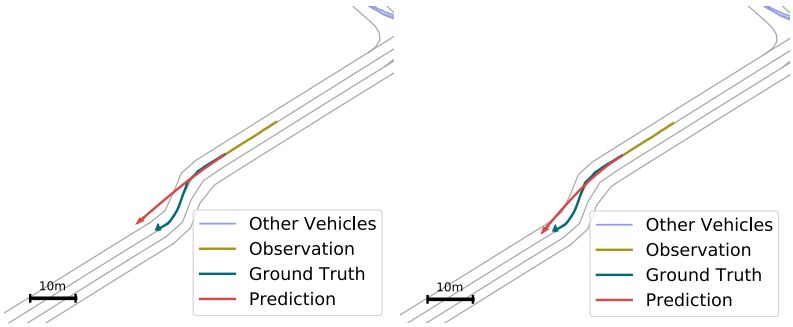


Figure 3.5: The output of the original model (the left) vs the robust model (the right) in a generated scene. While the original model has a trajectory in non-drivable area, the robust model predicts without any off-road.

fine-tuned model is less vulnerable to the generated scenes by predicting 40% less SOR and 30% less HOR in the Full setting. While the results show improvements in all transformation powers, the gains in extreme cases are higher, *i.e.*, the model can handle them better after fine-tuning.

In Figure 3.5, the prediction of the original model is compared with the prediction of the robust model. The original model cannot predict without off-road while the fine-tuned model is able to predict reasonable and without any off-road point.

3.4.5 Discussions

In this section, we perform experiments and bring speculations to shed light on the weaknesses of the models.

1. We study the ability to transfer the generated scenes to new models, *i.e.,* how models perform on the scenes generated for other models. We conduct this experiment by storing the generated scenes for a source model which lead to an off-road prediction, and evaluate the performance of target models on the stored scenes. Table 3.4 shows that the transferred

Source models	Target models		
	LaneGCN	DATF	WIMP
LaneGCN	34 / 100	37 / 82	20 / 61
DATF	11 / 44	52 / 100	13 / 46
WIMP	20 / 63	40 / 82	36 / 100

Table 3.4: Studying the transferability of the generated scenes. We generate scenes for source model and keep the ones that have off-road prediction by the source model. The target models are evaluated using those scenes. The reported numbers are SOR/HOR values. Numbers are rounded to the nearest integer.

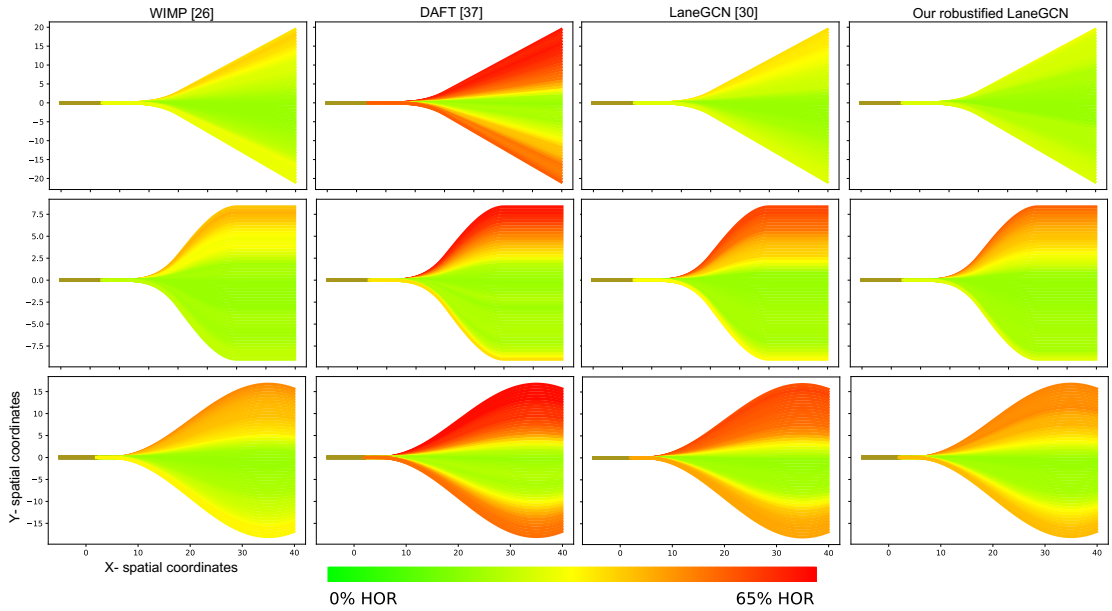
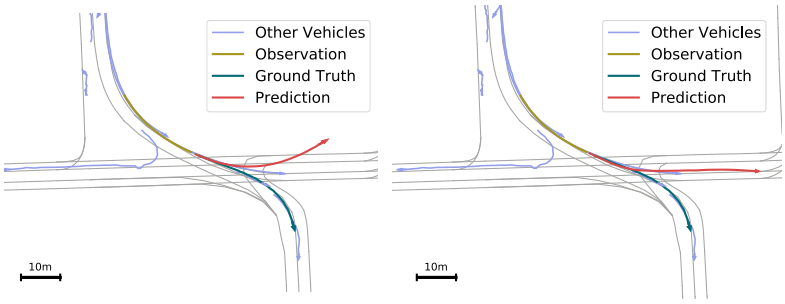


Figure 3.6: The qualitative results of baselines for different transformation functions. The red color indicates more off-road prediction in those scenes and the green indicates higher admissible ones. Usually the models fail in turns with high curvature. We could successfully make the LaneGCN model more robust by fine-tuning.

scenes are still difficult cases for other models.

2. We study how models perform with smoothly changing the transformation functions parameters. To this end, we smoothly change the transformation parameters for 100 random scenes and visualize the heatmap of HOR for the generated scenes. Figure 3.6 demonstrates that models are more vulnerable to larger transformation parameters, *i.e.*, sharper turns. Also, it shows more off-road in the left turns compared with the right ones which could be due to the biases in the dataset [75]. A clear improvement is visible in the robust model.

3. Our experiments showed that while the model has almost zero off-road rate in the original



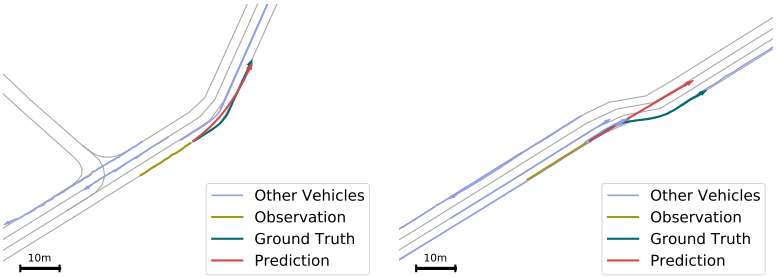
(a) Before robustness (b) After robustness

Figure 3.7: The output of the model before and after the robustness in a sample which requires reasoning over the scene. We observe that the model before robustness mainly uses h to predict instead of reasoning over the scene. However, after robustness, it reasons more over the scene.

scenes, it suffers from over 60% off-road rate in the generated ones. In order to hypothesize the causes of this gap, we explored the training data. We observed that in most samples, the history h has enough information about the future trajectory which reduces the need for the scene reasoning. However, our scene generation approach changes the scene such that h includes almost no information about the future trajectory. This essentially makes a situation that requires scene reasoning. We speculate that this feature is one factor that makes the generated scenes challenging. Note that this does not contradict the ablations in [15] as their performance measure is accuracy. Figure 3.7a shows a failure of the model where the prediction is only based on h instead of reasoning over the scene. However, the robust model learned to reason over the scene, as shown in Figure 3.7b. While our discussion is an observational hypothesis, we leave further studies for future works.

4. In some cases, our generated scene could not lead to an off-road prediction. One such example is depicted in Figure 3.8a.

5. While our method offers a new approach for assessing trajectory forecasting models, it has some limitations. First, our transformation functions are limited, and they cannot cover all real-world cases. We, however, propose a general methodology that can be expanded by adding other types of transformations. To demonstrate it, we add lane merging to the framework, which causes 14% HOR. Second, in addition to the off-road criterion, there exist other failure criteria. For instance, collision with other agents or abnormal behaviors like sudden lane changes. By choosing collision with other agents as criterion, HOR becomes 1.68% in the generated scenes while it is 0.55% in original data. Moreover, Figure 3.8b shows one scenario in which the predictions of the model are in the drivable area but the sudden lane change is abnormal.



(a) (b)

Figure 3.8: Some successful cases of the prediction model. In (a), the model follows the road and predicts without any off-road. In (b), while the model predicts on-road, it suddenly changes its lane.

3.5 Conclusions

We presented a conditional scene generation method. We showed that several state-of-the-art trajectory forecasting models fail in our generated scenes. Notably, they have high off-road rate in their predictions. Next, leveraging image retrieval techniques, we retrieved real-world locations that partially resemble the generated scenes and demonstrate their failure in those locations. We made the model robust against the generated scenes. We hope that this framework helps to better evaluate the prediction models which are involved in the autonomous driving systems.

In these two chapters, we evaluated trajectory forecasting models that integrate scene considerations and interpersonal interactions, revealing their shared weaknesses and limitations. In the next chapter, we want to engineer a robust architecture for trajectory forecasting that adapts to a wide range of scenarios.

4 Trajectory Forecasting using Visual Cues

This chapter is based on the following articles:

Saeed Saadatnejad, Yang Gao, Kaouther Messaoud and Alexandre Alahi, *Social-Transmotion: Promptable Human Trajectory Prediction*, under review, 2023

The code and a summary video of the work can be found on the project's webpage[1] and the earlier project's[2].

4.1 Introduction

In the preceding chapters, we assessed the performance of trajectory forecasting models, uncovering their inherent weaknesses and limitations. Guided by these findings, this chapter presents a novel and versatile forecasting model that seeks to overcome these challenges by integrating additional inputs.

Trajectory forecasting models aim to forecast the future positions of objects or people based on a sequence of observed 3D positions in the past. Despite acknowledging the inherent stochasticity that arises from human free will, traditional predictors have limited performance, as they typically rely on a single data point per person (i.e., their x-y coordinates on the ground) as input. This singular focus neglects a wealth of additional signals, such as body language, social interactions, and gaze directions, that humans naturally exhibit to communicate their intended trajectories.

In this study, we explore the signals that humans consciously or subconsciously use to convey their mobility patterns. For example, individuals may turn their heads and shoulders before altering their walking direction—a visual cue that cannot be captured using a sequence of spatial locations over time. Similarly, social interactions may be anticipated through gestures like hand waves or changes in head direction. Our goal is to propose a generic architecture for

[1] https://github.com/vita-epfl/social-transmotion/
[2] https://github.com/vita-epfl/bounding-box-prediction

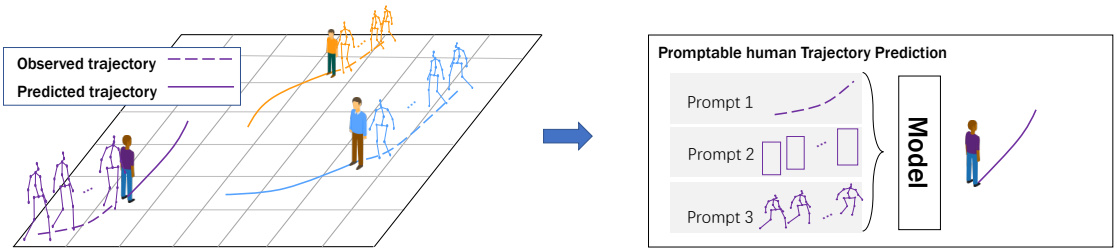


Figure 4.1: We present the task of *promptable human trajectory forecasting*: Predict human trajectories given any available prompt such as past trajectories or body poses of all pedestrians. Our model dynamically assesses the significance of distinct visual cues of both the primary and neighboring pedestrians and predicts more accurate trajectories.

human trajectory forecasting that leverages additional information whenever they are available (e.g., the body poses). We incorporate the sequence of observed cues as input, along with the observed trajectories, to predict future trajectories, as depicted in Figure 4.1. We translate the idea of a prompt from NLP to the task of human trajectory forecasting, where a prompt can be a sequence of x-y coordinates on the ground, bounding boxes or body poses. We refer to our task as *promptable human trajectory forecasting*. We embrace the multi-modal nature of human behavior by accommodating various visual cues to better capture the intricacies and nuances of human motion, leading to more accurate trajectory forecasts. The challenge lies in effectively encoding and integrating all these visual cues into the prediction model.

We introduce *Social-Transmotion*, a universal and adaptable transformer-based model for human trajectory forecasting. This model seamlessly integrates various types and quantities of visual cues, thus enhancing adaptability to diverse data modalities and exploiting rich information for improved prediction performance. Social-Transmotion processes a sequence of observed trajectories alongside corresponding visual cues of pedestrians within a scene. Its dual-transformer architecture dynamically assesses the significance of distinct visual cues of both the primary and neighboring pedestrians, effectively capturing relevant social interactions and body language cues. To ensure the generality of our network, we employ a training strategy that includes selective masking of different types and quantities of visual cues. Our model exhibits robustness and resilience, maintaining operational functionality even in the absence of certain visual cues, such as relying on bounding boxes when pose information is unavailable, or using only trajectory inputs when no visual cues are accessible.

Our experimental results demonstrate that Social-Transmotion outperforms baseline models in terms of accuracy. Additionally, we provide a comprehensive analysis of the usefulness of different visual representations, including 2D and 3D body pose keypoints and bounding boxes, for trajectory forecasting. We show that 3D pose keypoints more effectively capture social interactions, while 2D pose keypoints can be a good alternative when 3D pose information is unavailable. We also consider the requirements for using poses from all humans at all times and the necessity of 3D versus 2D poses or even just bounding boxes. In some applications, only the latter may be available. We provide an in-depth analysis of these factors in Section 4.4.

In summary, our contributions are twofold. First, we present Social-Transmotion, a generic Transformer-based model for promptable human trajectory forecasting, designed to flexibly utilize various visual cues for improved accuracy, even in the absence of certain cues. Second, we provide an in-depth analysis of the usefulness of different visual representations for trajectory forecasting. The code for our proposed model will be made publicly available upon publication.

4.2 Related Work

4.2.1 Attention-based Human Trajectory Forecasting

Human trajectory forecasting has evolved significantly over the years. Early models, such as the Social Force model, focused on the attractive and repulsive forces among pedestrians [118]. Later, Bayesian Inference was employed to model human-environment interactions for trajectory forecasting [119]. As the field progressed, data-driven methods gained prominence [14], [120]–[126], with many studies constructing human-human interactions [14], [120], [123], [125] to improve predictions. For example, using hidden states to model observed neighbor interactions [120], or the directional grid for better social interaction modeling [14]. In recent years, researchers have expanded the scope of social interactions to encompass human-context interactions [119], [124] and human-vehicle interactions [127], [128]. Moreover, multimodality has been effectively modeled using various techniques, such as generative adversarial networks (GANs) [47], [121], [129], [130], recurrent neural networks (RNNs) [43], [131], and diffusion models [132].

The introduction of Transformers and positional encoding [133] has led to their adoption in sequence modeling, owing to their capacity to capture long-range dependencies. This approach has been widely utilized recently in trajectory forecasting [45], [122], [134]–[137]. Despite advancements in social-interaction modeling, previous works have predominantly relied on sequences of pedestrian x-y coordinates as input features. With the advent of datasets providing more visual cues [138]–[140], more detailed information about pedestrian motion is now available. Therefore, we design a generic transformer that can benefit from incorporating visual cues in human trajectory forecasting.

4.2.2 Visual Cues for Trajectory Forecasting

Multi-task learning has emerged as an effective approach for sharing representations and leveraging complementary information across related tasks. Numerous pioneering studies have demonstrated the potential benefits of incorporating additional associated tasks into human trajectory forecasting, such as intention prediction [141], 2D/3D bounding-box prediction [142], and action recognition [143].

The human pose serves as a potent indicator of human intentions. Owing to the advancements in pose estimation [144], 2D poses can now be readily extracted from images. In recent years, a couple of studies have explored the use of 2D body pose as visual cues for trajectory forecasting

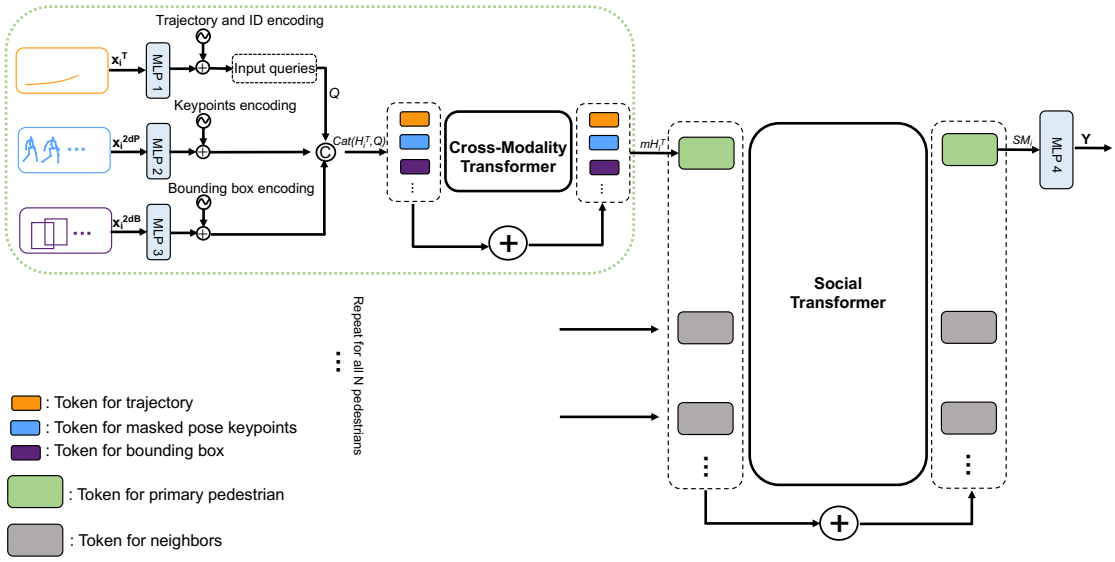


Figure 4.2: Social-Transmotion: A Transformer-based model that integrates visual cues, specifically 3D human poses, to enhance the accuracy and social-awareness of human trajectory forecasting.

in image/pixel space [145], [146]. However, our work concentrates on trajectory forecasting in camera/world coordinates, which offers more extensive practical applications. Employing 2D body pose presents limitations, such as information loss in depth, making it difficult to capture the spatial distance between agents. In contrast, 3D pose circumvent this issue and have been widely referred to in pose estimation [147], pose forecasting [148]–[150], and pose tracking [151]. Nevertheless, 3D pose data may not always be available in real-world scenarios. Drawing inspiration from the improved intention prediction achieved through the utilization of bounding boxes [141], we have also included this visual cue in our exploration. Our goal is to investigate the effects of various visual cues, including but not limited to 3D human pose, on trajectory forecasting.

A study with close ties to our research emphasizes the significance of an individual pedestrian's 3D body pose in predicting their trajectory [152]. However, our research incorporates social interactions among poses, a feature overlooked in their study. Furthermore, in contrast to another research that focused solely on head orientation as a feature [153], we explore more granular representations. Our work not only considers the effect of social interactions between 3D pose but also other visual cues, amplifying trajectory forecasting precision. Moreover, our adaptable network is capable of harnessing any available visual cues.

4.3 Method

Our main objective is to tackle the task of predicting future global trajectory coordinates. We denote the observed time-steps as $t = 1, ..., T_{obs}$ and the prediction time-steps as $t = T_{obs} +$

$1, ..., T_{pred}$. In order to enhance the prediction performance, we develop an adaptable model that effectively utilizes various visual cues alongside the historical trajectory information. We also recognize that different scenarios may provide varying numbers of visual cues, and thus, our model is designed to accommodate these potential variations. To achieve this, we incorporate additional visual cues, such as 2D or 3D human pose information and bounding boxes, into our trajectory forecasting models and we train our model to be adaptable to different types and numbers of cues.

Our model has a generic architecture capable of processing different cue types and numbers, allowing to capture distinct and relevant information. As depicted in Figure 4.2, it consists of two transformers, each fulfilling specific roles in information capture. The cross-modality transformer receives inputs such as the past 2D coordinates of the agent. It can also leverage additional cues to enhance the motion representation, such as the agent's 2D or 3D pose information and bounding boxes for past time-steps. By incorporating these diverse cues, the cross-modality transformer can encode a more informative representation of the agent's behavior. The social transformer is responsible for integrating the outputs from the first transformers of different agents. By combining these individual representations, the social transformer captures the interactions between agents, allowing the model to analyze the interplay and dependencies between them.

4.3.1 Problem Formulation

We denote the global trajectory sequence of pedestrian i as $\mathbf{x_i^T}$, the 3D and 2D local pose coordinates as $\mathbf{x_i^{3dP}}$ and $\mathbf{x_i^{2dP}}$ respectively, and the 3D and 2D bounding box coordinates as $\mathbf{x_i^{3dB}}$ and $\mathbf{x_i^{2dB}}$, respectively. In a scene with N pedestrians, the complete network input is $\mathbf{X} = [X_1, X_2, X_3, ..., X_N]$, where $X_i = \{\mathbf{x_i^c}, \mathbf{c} \in \{\mathbf{T}, \mathbf{3dP}, \mathbf{2dP}, \mathbf{3dB}, \mathbf{2dB}\}\}$ depending on the availability of different cues. The tensor $\mathbf{x_i^c}$ has a shape of (T_{obs}, e^c, f^c), where e^c represents the number of elements in a specific cue (for example the number of keypoints) and f^c denotes the number of features for each element.

Without loss of generality, we consider X_1 as the primary agent. The network output, $\mathbf{Y} = Y_1$, contains the predicted future trajectory of the primary pedestrian, following the standard notation.

4.3.2 Input Cues Embeddings

To effectively incorporate the visual cues into our model, we employ a cue-specific embedding layer to embed the coordinates of the trajectory and all visual cues at each time step. In addition, we utilize positional encoding techniques to represent the input cues' temporal order. We also encounter two additional aspects that need to be encoded: the identity of the person associated with each cue and the keypoint type for keypoint-related cues (e.g., neck, hip, shoulder). To tackle this, we introduce three distinct embeddings: one for temporal order, one for person identity, and one for keypoint type. The temporal order embedding facilitates the understanding of the sequence of cues, enabling the model to capture temporal dependencies and patterns. The person

identity embedding allows the model to distinguish between different individuals within the input data. Lastly, the keypoint type embedding enhances the model's ability to extract relevant features and characteristics associated with different keypoint types movement. These embeddings are randomly initialized and learned during the training process.

$$H_i^c = MLP^c(\mathbf{x_i^c}) + P,$$

The resulting tensor H_i^c has a shape of (T_{obs}, e^c, D), where D represents the embedding dimension, MLP^c refers to cue-specific MLP embedding layers, and the tensor P contains positional encoding information.

4.3.3 Cross-Modality Transformer (CMT)

The CMT in our model is designed to process various inputs embedding vectors, including the past 2D coordinates of the agent, its 2D or 3D pose information and 2D or 3D bounding boxes from past time-steps. By incorporating these different cues, the CMT is capable of encoding a more comprehensive and informative representation of the agent's motion dynamics.

In addition, our CMT receives a set of latent queries Q. At the final layers of the network, each latent query is projected to correspond to one of the potential future positions of the target agent. By incorporating these latent queries, the CMT gains the capability to attend to specific future positions and encode relevant information accordingly. This mechanism enables our model to generate predictions and anticipate the potential trajectories of the target agent.

Furthermore, this CMT employs shared parameters to process the various modalities and ensure efficient information encoding across different inputs. It effectively encodes information from different cues, facilitating a richer understanding of the agent's dynamics.

$$mH_i^T, mH_i^c = \mathbf{CMT}(concat(H_i^T, Q), H_i^c, c \in \{3dP, 2dP, 3dB, 2dB\}).$$

$\mathbf{CMT}$ transforms the latent representation of agent motion, $concat(H_i^T, Q)$, into a motion cross-modal tensor mH_i^T with shape (T_{pred}, e^T, D). Similarly, each cues embedding tensor H_i^c is mapped to mH_i^c with shape (T_{obs}, e^c, D).

It is important to note that while our CMT receives inputs from various cues, only the motion cross-modal tensor is passed to the second transformer. This decision is based on the assumption that these motion cross-modal features capture and encode information from the different cues.

4.3.4 Social Transformer (ST)

ST in our model integrates the outputs from the CMT of all agents. This integration process captures the interactions between agents, allowing the model to analyze the interplay and dependencies that exist among them. By combining the individual representations from different agents, the ST creates a comprehensive representation of the collective behavior, considering the influence and interactions among the agents. This enables the model to better understand and predict the complex dynamics in multi-agent scenarios.

$$SM_i = ST(mH_i^T, i \in [1, N]).$$

The Social Transformer ST transforms the motion cross-modal tensor of each agent H_i^T to a socially aware encoding tensor SM_i with shape (T_{pred}, e^T, D).

Finally, the output latent queries of the primary agent generated by the social transformer in SM_1 undergo a projection layer. This projection transforms it into the 2D coordinate predictions of the future positions of the primary agent.

4.3.5 Training Procedure

To ensure the generality and adaptability of our network, we employ a training approach that involves masking different types and quantities of visual cues. Each sample in the training dataset is augmented with a variable combination of cues, including trajectories, 2D or 3D human pose information, and bounding boxes. This masking technique enables our network to learn and adapt to various cue configurations during training.

Subsequently, we conduct testing to evaluate the model's performance across different combinations of visual cues. By systematically varying the presence or absence of specific cues in the input, we assess the model's ability to leverage different cues for accurate trajectory forecasting.

By training and testing the network with diverse sets of cues, we ensure that our model is robust and adaptable to a wide range of scenarios. This flexibility allows it to handle situations where certain cues may be unavailable or less informative, ultimately enhancing its overall performance in trajectory forecasting tasks.

Our model is trained with a Mean Square Error (MSE) loss function between $\mathbf{Y}$ and its ground truth $\hat{\mathbf{Y}}$.

4.4 Experiments

In this section, we present the experimental setup, including the datasets used, evaluation metrics, and an extensive analysis of the results in both quantitative and qualitative aspects.

4.4.1 Datasets

We evaluate our models on three publicly available datasets: JTA [138], Pedestrians and Cyclists in Road Traffic [152], and a newly introduced dataset named JRDB [139].

1. The JTA dataset is a large-scale synthetic dataset containing 256 training sequences, 128 validation sequences, and 128 test sequences, with a total of approximately 10 million 3D keypoints annotations. The abundance of data and multi-agent scenarios in this dataset enables a thorough exploration of our models' potential performance. We used TrajNet++ benchmark [14] to pre-process this dataset. We predict the location of future 12 time-steps given the previous 9 time-steps under 2.5 fps.

2. The JRDB dataset is a real-world dataset that provides a diverse set of pedestrian trajectories and 2D bounding boxes, allowing for a comprehensive evaluation of our models in both indoor and outdoor scenarios. Specifically, we used '$gates - ai - lab - 2019 - 02 - 08_0$' for validation, indoor scenario '$packard - poster - session - 2019 - 03 - 20_1$' and outdoor scenario '$tressider - 2019 - 03 - 16_0$' for evaluation and the other static scenarios for training. Similar to JTA, we predict future 12 time-steps given past 9 time-steps under 2.5 fps.

3. The Pedestrians and Cyclists in Road Traffic dataset, is a real-world dataset containing more than $2,000$ trajectories of pedestrians with 3D body poses recorded in urban traffic environments. It is specifically designed for single-person scenarios in urban traffic and has gained attention for research in autonomous driving. Leveraging a sliding window approach, we create a substantial test set of $50,000$ samples, enabling comprehensive evaluations of our model's performance. In line with [152], we adopted the same experimental setup, allowing 1 second for observation and forecasting the future 2.52 seconds under 25 fps.

4.4.2 Metrics and Baselines

We evaluate our models using the TrajNet++ benchmark [14], which includes ADE and FDE metrics. Additionally, we use ASWAEE [152] to assess trajectory forecast at specific time horizons. In summary, we employ the following metrics:

1. ADE - The average L2 displacement error between predicted and actual pedestrian locations across all prediction time-steps;

2. FDE - The L2 displacement error between the predicted and actual pedestrian locations at the final prediction time-step;

3. ASWAEE - Average Specific Weighted Average Euclidean Error, which calculates the average displacement error per second for specific time points. Following [152], we compute it for five timeframes: [t=0.44s, t=0.96s, t=1.48s, t=2.00s, t=2.52s]

We selected the best-performing trajectory forecasting models [14], [120]–[122] from TrajNet++ leaderboard [14] that used different networks (LSTMs, Transformers, GANs) in addition to three recent high-performing models [131], [136], [137]. Moreover, we compared with another pose-based trajectory forecasting model [152].

4.4.3 Results

Quantitative Results

Table 4.1 presents the quantitative results of our experiments, comparing the baseline models with our proposed visual-cues-based models. The inclusion of pose information in our models significantly improves the accuracy of trajectory forecasting, as indicated by the enhanced ADE/FDE metrics. One possible explanation for this improvement is that pose-based models can capture body rotation prior to changes in walking direction.

3D pose yields better improvements compared to 2D pose. This can be attributed to the fact that modeling social interactions requires more spatial information, and 3D pose provides the advantage of depth perception compared to 2D pose.

The absence of pose information in the JRDB dataset led us to rely on bounding boxes as a visual cue instead. Results show that incorporating bounding boxes have a performance comparable to trajectory-based predictions alone. Additionally, we conducted a similar experiment on the JTA dataset and observed that the inclusion of 2D bounding boxes, in addition to trajectories, improved the FDE metric. However, it is important to note that the performance was still lower compared to utilizing 3D pose cues.

Furthermore, we conducted an experiment involving the use of trajectory, 3D pose and 3D bounding box. The results indicated that the performance of this combination was comparable to using trajectories and 3D poses alone. This suggests that, on average, incorporating 3D bounding boxes does not provide additional information beyond what is already captured by 3D poses. Lastly, we assessed the model's performance using all accessible cues: trajectory, 3D and 2D poses, and 3D and 2D bounding boxes, and it yielded similar results.

Qualitative Results

Figure 4.3 provides a visual comparison between Social-Transmotion, which uses only trajectory inputs, with its pose-based counterpart. The inclusion of pose information helps the model predict when and where the primary pedestrian will change direction and avoid collisions with neighbors. For instance, in the middle figure, adding pose enables the model to understand body rotation and collision avoidance simultaneously, resulting in a prediction closer to the ground truth.

Predicting sudden turns presents a significant challenge for trajectory forecasting models. However, the addition of pose information can help overcome this. As demonstrated in the middle

Models	Input Modality	JTA dataset		JRDB dataset	
		ADE	FDE	ADE	FDE
Social-GAN* [121]	T	1.66	3.76	0.50	0.99
Transformer [122]	T	1.56	3.54	0.56	1.10
Vanilla-LSTM [120]	T	1.44	3.25	0.42	0.83
Occupancy-LSTM [120]	T	1.41	3.15	0.43	0.85
Directional-LSTM [14]	T	1.37	3.06	0.45	0.87
Dir-social-LSTM [14]	T	1.23	2.59	0.48	0.95
Social-LSTM [120]	T	1.21	2.54	0.47	0.95
Autobots [136]	T	1.20	2.70	0.42	0.79
Trajectron++ [131]	T	1.18	2.53	0.42	0.79
EqMotion [137]	T	1.13	2.39	0.42	0.80
Social-Transmotion	T	1.00	2.02	0.40	0.77
Social-Transmotion	T + 2d P	0.98	1.97	/	/
Social-Transmotion	T + 3d P	0.92	1.87	/	/
Social-Transmotion	T + 2d BB	0.97	1.92	**0.38**	**0.75**
Social-Transmotion	T + 3d P + 3d BB	0.92	1.87	/	/
Social-Transmotion	T + 3d P + 2d P + 3d BB + 2d BB	**0.91**	**1.85**	/	/

Table 4.1: Quantitative results on the JTA and JRDB datasets. The unit for ADE, FDE is meter. 'T' refers to Trajectory, 'P' signifies Pose, and 'BB' denotes Bounding Box.

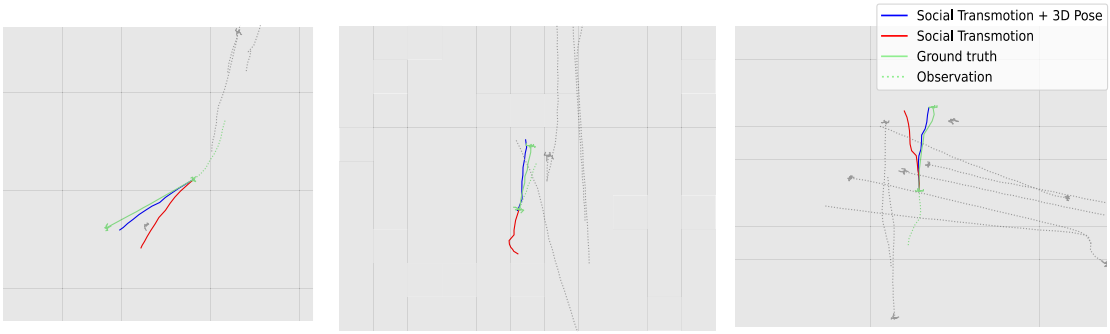


Figure 4.3: Qualitative examples on JTA dataset [138]. Each individual example shows the trajectories of pedestrians for a specific scene. For the primary agent, the ground truth, models' prediction and their pose-based counterparts' are in green, red, and blue, respectively. All other agents are in gray. We have also visualized the pose of the last observed frame as it can reflect the walking direction and body rotation.

figure, the pose-based model excels in scenarios involving sudden turns, leveraging pose to anticipate forthcoming changes in walking state, an aspect the conventional model fails to capture.

Figure 4.4 shows some failure cases of Spocial-Transmotion. The first one shows an inevitable scenario where the individual alters their path in the middle of the future horizon. The second figure shows a crowded context where prediction is challenging, and the third one shows a situation where pose information offers limited value. The integration of supplementary visual

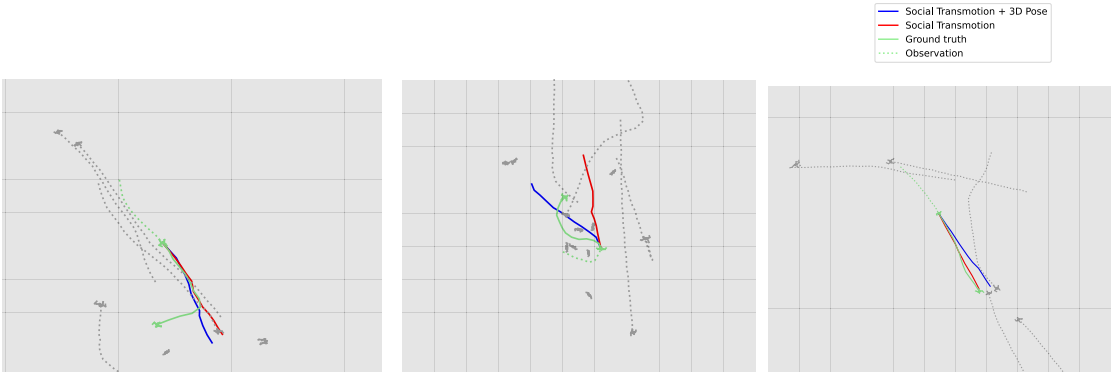


Figure 4.4: Some qualitative failure cases. The efficacy of pose information varies; the left figure demonstrates an inevitable scenario where the individual alters their path in the middle of the future horizon. The middle figure shows a crowded context, and the right one shows a situation where pose information offers limited value.

cues like gaze or the original scene image could potentially offer advantageous improvements.

4.4.4 Discussions

What if we use one single transformer instead of dual transformers?

In our Social-Transmotion architecture, we initially designed two transformers: one for individual pedestrian feature extraction and another for capturing pedestrian interactions. In this study, we compared this dual-transformer setup with a unified single-transformer model. In the single-transformer configuration, we combined the tokens of all pedestrian features, allowing attention to both different features and different pedestrians simultaneously. However, as shown in Table 4.2, there was a notable decrease in performance when the model attended to all pedestrians' features concurrently. This highlights the effectiveness of our original architecture in separating the primary pedestrian's representations from social interactions.

Additionally, we conducted an experiment to assess the significance of social modeling by excluding the Social Transformer and focusing solely on the primary agent through the Cross-Modality Transformer. As indicated in Table 4.2, the performance significantly declined without the Social Transformer, emphasizing the importance of modeling social interactions between agents.

Finally, we tested swapping the order of ST and CMT, resulting in a significant 24% drop in ADE and a 25% decrease in FDE. This indicates that it is more effective to initially extract features of all agents (CMT) and then subsequently aggregate the features of all agents (ST). Our hypothesis is that the relationships between an individual's joints positions across different time steps hold greater significance compared to the interactions among joints of multiple individuals within a specific time step. The ST-first approach places a demanding task on the network, compelling it

Models	Input Modality	ADE	FDE
Social-Transmotion	T+3d P	0.92	1.87
Social-Transmotion (one single transformer)	T+3d P	1.06	2.13
Social-Transmotion (without ST)	T+3d P	1.13	2.32
Social-Transmotion	T+3d P	0.92	1.87
Social-Transmotion	T+3d P (only primary's pose)	0.95	1.96
Social-Transmotion	T+3d P (only last obs. pose)	0.96	1.94
Social-Transmotion	T+3d P (only head pose)	1.00	2.02

Table 4.2: Ablation studies of Social-Transmotion. The top section compares different architectural choices, and the bottom section explores variations in the use of 3D pose.

to extract useful information from numerous irrelevant connections.

What if we only use the primary pedestrian's poses?

In our study, we observed that our 3D Pose-based model achieves good performance. To delve deeper into the contribution of pose information in improving ADE and FDE, we conducted an ablation study. We specifically examined the impact of retaining only the primary pedestrian's pose while excluding neighboring poses. This allowed us to assess whether the performance improvement was solely attributable to the primary pedestrian's pose or if pose interactions played a role. Results in Table 4.2 indicate that using only the primary pedestrian's pose leads to better performance compared to the baseline model. However, it is important to note that incorporating all pedestrian poses yields even greater improvements which highlights the importance of considering pose interactions in trajectory forecasting.

What if we only input the last observed pose?

In a further ablation study investigating the influence of pose in our model, we utilized only the last observed pose frame as a visual cue for all agents in the scene. The results in Table 4.2 indicate a performance comparable to when all observed frames were incorporated. This suggests that trajectory forecasting relies on the last observation frame more than other frames for accurate predictions.

What if we use only head pose?

Our investigation also extended to the exclusive use of head pose as a visual cue, meaning all non-head pose keypoints were excluded. As shown in Table 4.2, the performance achieved when only utilizing the head pose is analogous to the trajectory-only model. This outcome indicates the necessity of incorporating other pose data for enhanced model performance.

To explore the impact of different keypoints/frames on trajectory forecasting accuracy, we

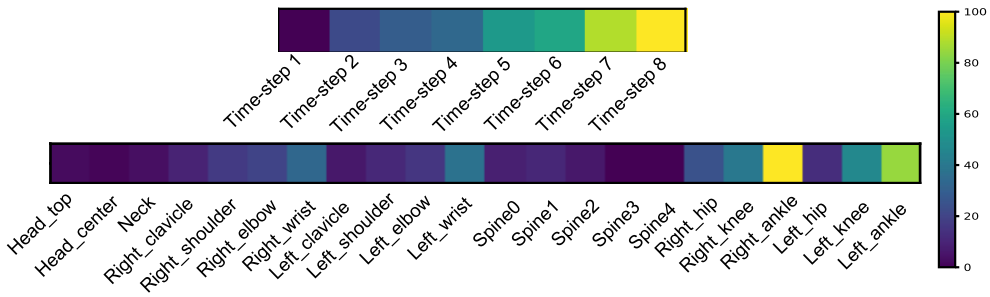


Figure 4.5: Depiction of attention maps: temporal (top) and spatial (bottom), highlighting the importance of specific time frames and body keypoints in trajectory forecasting.

generated attention maps as shown in Figure 4.5. The first map illustrates temporal attention, while the second map represents spatial attention. The attention weights assigned to earlier frames are comparatively lower, indicating that later frames contain more valuable information for trajectory forecasting. In simpler scenarios, even the last observed frame may be sufficient, as demonstrated in our previous ablation study. However, in more complex scenarios, a larger number of observation frames may be required. We also observed that specific keypoints, such as the ankles, wrists, and knees, play a significant role in determining direction and movement. Generally, there is symmetry across different body points, with a slight tendency towards the right. We hypothesize it may be attributed to data bias. These findings open up opportunities for further research, particularly in identifying a sparse set of essential keypoints that can offer advantages in specific applications.

Robustness against imperfect observations

In real-world situations, obtaining complete observations can be challenging due to occlusions or low confidence in pose detection. In Chapter 2, we observed models are vulnerable to small perturbations. Our interest here lies in examining the impact of small perturbations on the Social-Transmotion model.

We investigate the model's performance under varying percentages of masking during inference. It is important to note that if a model is trained on complete input (without masking) and then evaluated with 90% available trajectory (only 10% missing), the ADE/FDE metrics drop by 53% and 49%, respectively. However, our model, empowered by the masking strategy, exhibits a marginal decline in performance, as indicated in Table 4.3, corresponding to much larger missing percentages of input during test time.

We further investigated the model's robustness by introducing various noise patterns in 3d pose. The results in the bottom part of Table 4.3, consistently indicate the model's robust performance in scenarios with missing whole poses in some frames, structural missing poses, and the addition of Gaussian noise to keypoints.

Model	Input Modality	ADE/FDE
Social-Transmotion	100% T + 100% 3d P	0.92/1.87
	90% T + 90% 3d P	0.92/1.87
	50% T + 50% 3d P	1.05/2.10
	50% T + 10% 3d P	1.13/2.23
Social-Transmotion	T + 50% Missing Arm and Leg Keypoints in 3d P	0.93/1.89
	T + Right Leg Missing 3d P (in all frames)	0.92/1.88
	T + Whole Frame Missing 3d P (with 25% probability)	0.92/1.88
	T + 3d P with Random Gaussian Noise (N(0, 25))	0.99/1.98
	T + 3d P with Random Gaussian Noise (N(0, 50))	1.05/2.08

Table 4.3: Robustness evaluation of Social-Transmotion given imperfect pose and trajectories.

Models	ASWAEE
c_{traj} [152]	0.57
d_{traj} [152]	0.60
$c_{traj,pose}$ [152]	0.51
$d_{traj,pose}$ [152]	0.56
Social-Transmotion	0.48
Social-Transmotion + 3d P	**0.40**

Table 4.4: Quantitative results on the Pedestrians and Cyclists in Road Traffic dataset [152]. The models are compared using ASWAEE metric, measured in meters, with lower values denoting lower displacement error.

Computational costs

In inference time, Social-Transmotion showcases a prediction time of 8.1 milliseconds on average for forecasting future timesteps (4.8 seconds) while utilizing all visual cues. Notably, EqMotion requires 11.1 ms, and Autobots requires 13.4 ms for the same task, underscoring the efficiency of our model's forward process in comparison. However, it is important to acknowledge that our approach relies on additional estimation methods within its pipeline. Hence, when considering the complete processing time, especially for potential real-world deployments, these factors should be taken into account. These timings were recorded using a single 32 GB NVIDIA V100 GPU.

4.4.5 Experiment on Pedestrians and Cyclists in Road Traffic Dataset

Table 4.4 compares the performance of our proposed model, Social-Transmotion, with the previous work that utilized trajectory or 3D pose for human trajectory forecasting [152]. Here, the notations 'c' and 'd' represent two variations of their model using a continuous or discrete approach, respectively. The results indicate the effectiveness of our architecture and its proficiency in utilizing pose information to enhance prediction accuracy.

4.5 Conclusions

In this chapter, we introduced Social-Transmotion, a generic Transformer-based model adept at managing diverse visual cues in varying quantities, thereby augmenting trajectory data for enhanced human trajectory forecasting. A series of rigorous experiments underscored the value of integrating visual cues, such as human body pose, into the trajectory forecasting framework. By embracing the multi-modal aspects of human behavior, our approach pushed the limits of conventional trajectory forecasting performance.

While our model can work with any visual cue, we have examined a limited set of visual cues and noted instances where they did not consistently enhance trajectory prediction performance. In the future, one can study the potential of alternative visual cues such as gaze direction, actions, and other attributes, considering their presence in datasets. Although our model demonstrates strong performance even without visual cues, it is important to note that we rely on estimation methods to derive these cues. An intriguing avenue for research involves benefiting directly from images by developing efficient feature extraction networks. These networks could facilitate the transformation of images into optimized prompts, enabling the direct utilization of visual information.

Having delved into trajectory forecasting in previous chapters, our exploration naturally progresses to a more granular level. The next two chapters extend our focus to the finer details of motion - pose forecasting.

5 Human Pose Forecasting with Uncertainty

This chapter is based on the article:

Saeed Saadatnejad, Mehrshad Mirmohammadi, Matin Daghyani, Parham Saremi, Yashar Zoroofchi Benisi, Amirhossein Alimohammadi, Zahra Tehraninasab, Taylor Mordan and Alexandre Alahi, *Toward Reliable Human Pose Forecasting with Uncertainty*, under review, 2023

The code and a summary video of the work can be found on the project's webpage[1] and the earlier project's[2].

5.1 Introduction

Building upon our insights from trajectory forecasting in previous chapters, we now turn to the next layer of complexity in this thesis: pose forecasting.

Robots and humans are poised to work in close proximity. Yet, current technology struggles to read and anticipate the motion dynamics of humans. Predicting human poses enables a safe co-existence between humans and robots, with direct applications in autonomous driving [154], human-robot collaboration [155], robot navigation [156], and healthcare [157] The task of human pose forecasting consists in predicting a sequence of future 3D poses of a person, given a sequence of past observed ones. The field is now witnessing an arms race of forecasting models using different architectures that have shown increasing performances [158]–[160].

Forecasting human poses is a difficult problem with multiple challenges to solve: it mixes both spatial and temporal reasoning, with a huge variability in scenarios; and human behavior is difficult to predict, as it changes in dynamic and multi-modal ways to react to its environment. To guarantee safe interactions with humans, robots should not only predict human motions, but also identify scenarios in which they are uncertain [161]–[164], and act accordingly. Figure 5.1

[1] https://github.com/vita-epfl/unposed
[2] https://github.com/vita-epfl/decoupled-pose-prediction

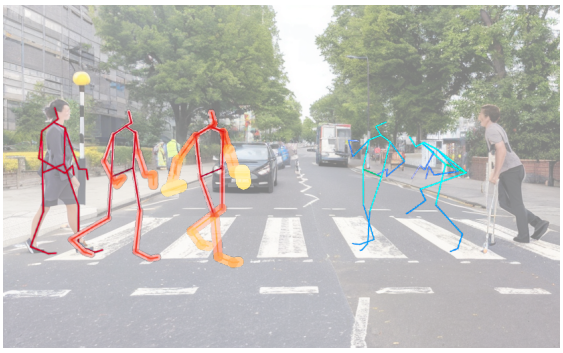


Figure 5.1: We propose to model two kinds of uncertainty: 1) aleatoric uncertainty, learned by our model to capture the temporal evolution of uncertainty, which becomes more prominent over time, as depicted by the lighter colors and thicker bones for the left person; 2) epistemic uncertainty to detect out-of-distribution forecast poses coming from unseen scenarios in training, such as for the right person.

illustrates a pose forecasting scenario. Without an uncertainty measure, all the forecast poses are considered valid. However, uncertainty measures can detect unconfident outputs and treat them with more caution. Recently, several works have shown the benefits of estimating uncertainty for classification [163], [164] and regression tasks [161], [162], but how to apply this principle to pose forecasting is not yet studied.

In this chapter, we present two solutions to capture the uncertainty of pose forecasting models from two important perspectives. The first one deals with the aleatoric uncertainty, *i.e.*, the irreducible intrinsic uncertainty in the data. We reformulate the pose forecasting objective function to capture the aleatoric uncertainty. To reduce the number of learned parameters and improve stability, we introduce uncertainty priors based on our knowledge about the uncertainty, *e.g.*, that the uncertainty increases with time. We then train the forecasting model with the new objective function. This allows the model to focus its capacity to learn forecasting at shorter time horizons, where uncertainty is lower and learning is more meaningful, compared to longer ones that are intrinsically much harder and uncertain to forecast.

The second one is about epistemic uncertainty which shows the model's lack of knowledge. To this end, we define a model-agnostic uncertainty metric as an indicator of the certainness and trustability of pose forecasting models in the real world. Unlike previous methods which require accessing model [165] (*i.e.*, white-box methods) or are specific to certain models [164], our approach does not require access to the model (*i.e.*, black-box approach) and is model-agnostic. Since there is no label for motions, we train a deep clustering network to learn the distribution of common poses and measure the dissimilarity between the predictions' embeddings and cluster centers. We apply our proposed uncertainty methods to several models from the literature and evaluate them on three well-known datasets (Human3.6M [140], AMASS [166], 3DPW [167]) and achieve up to 25 % improvement leveraging the aleatoric uncertainty and better performance in detecting out-of-distribution forecast poses using epistemic uncertainty.

Thanks to the large interest in pose forecasting, the field is advancing at a quick pace. However, this happens at the cost of unfair and non-unified evaluations. Concurrent papers all use disparate metrics and dataset setups to report their results, leading to ambiguities and errors in interpretation. In an effort to mitigate these discrepancies, we develop and release an open-source library for human pose forecasting named *unposed* [3]. This includes our re-implementations of over 10 models, processing codes for 5 widely-used datasets and 6 metrics, all implemented and tested in a standardized way, in order to ease the implementation of new ideas and promote research in this field. To summarize, our contributions are three-fold:

- We propose a method for incorporating priors to estimate the aleatoric uncertainty in human pose forecasting and demonstrate its efficacy in improving several state-of-the-art models on multiple datasets;
- We propose a model-agnostic metric of quantifying epistemic uncertainty to evaluate models in unseen situations, outperforming previous methods;
- We develop and publicly release an open-source library for human pose forecasting.

5.2 Related Work

5.2.1 Human Pose Forecasting

While the literature has extensively examined the forecasting of a sequence of future center positions at a coarse-grained level [14], [16] or a sequence of bounding boxes [141], [142], our focus in this chapter and the next chapter is on a more fine-grained forecasting *i.e.*, pose. Additionally, we limit our focus to the observation sequence alone, rather than incorporating context information [168], social interactions [149], or action class [169]. Many approaches have been proposed for human pose forecasting, with some using feed-forward networks [170] and many others using Recurrent Neural Networks (RNNs) to capture temporal dependencies [171], [172]. To better capture spatial dependencies of body poses, Graph Convolutional Networks (GCNs) have been utilized [158], [173], along with separating temporal and spatial convolution blocks and using trainable adjacency matrices [159], [174]. Attention-based approaches have also gained interest for modeling human motion, showing improvement with a spatio-temporal self-attention module [158]. More recently, forecasting in multiple stages [160] and a diffusion stochastic model with a transformer-based architecture [20] have been proposed. We can categorize all previous works into stochastic and deterministic models. Stochastic models [20], [175]–[181] can give diverse predictions, but in this chapter, we mainly focus on deterministic models [158]–[160], [182] as they provide more accurate predictions. Given the growing interest in this field, we believe that greater attention should be paid to uncertainty estimation in this task.

[3] https://github.com/vita-epfl/unposed

5.2.2 Uncertainty in Pose Forecasting

Knowing when a model does not know, *i.e.*, uncertain, is important to improve trustworthiness and safety [183]. Traditionally, uncertainty in deep learning is divided into data (aleatoric) and model (epistemic) uncertainty [161]. The aleatoric originates from the intrinsic noise and inherent uncertainty of data and cannot be reduced by improving the model, while the epistemic uncertainty shows the model's weakness in recognizing the underlying pattern of the data and can be reduced by enhancing the network architecture or increasing data. Many methods have been proposed to estimate and utilize these types of uncertainty in various tasks, including image classification [165], semantic segmentation [184], and natural language processing [185]. It has also been widely explored in pose estimation from images and videos [186]–[188], visual navigation and trajectory forecasting tasks [189]–[191] but not yet studied in human pose forecasting which includes spatio-temporal relationships modeling. We will show how modeling the uncertainty can improve accuracy.

Moreover, it is important to measure the epistemic uncertainty of models when going to be used in the real world. Bayesian Neural Networks (BNNs) have conventionally been used to formulate uncertainty by defining probability distributions over the model parameters [192]. However, the intractability of these distributions has led to the development of alternative approaches to perform approximate Bayesian inference for uncertainty estimation. One popular approach is Variational Inference [193], [194] because of their scalability. One important example is the technique of Monte Carlo (MC) dropout [165], which involves applying dropout [195] at inference time to model the parameters of the network as a mixture of multivariate Gaussian distributions with small variances. However, all those methods need to access the model and are not model-agnostic. Calibration [196] is another approach, but it requires the model to provide probabilities and deep neural networks have been shown to be poorly calibrated. One way to evaluate model reliability is by measuring the distance between a new sample and the training samples using a deep deterministic network, a technique that has been effective in image classification [163], [164]. However, this approach measures the uncertainty for their own model and is not applicable to measuring the uncertainty of different models. In contrast, Deep Ensembles [197] can measure the uncertainty of different models by training multiple neural networks independently and averaging their outputs at inference time. However, this method can be computationally expensive and slow. In this study, we concentrate on the model's output and define epistemic uncertainty as the extent to which the model's forecast resemble the training distribution. This way, we can measure the uncertainty in a black-box manner.

5.3 Aleatoric Uncertainty in Pose Forecasting

Pose forecasting models usually take as input a sequence x of 3D human poses with J joints in O observation time frames, and predict another sequence $\hat{y}$ of 3D poses to forecast its future y in the next T time frames. In addition to this, we also want a model to estimate its aleatoric uncertainty u along with the predicted poses $\hat{y}$, to indicate how reliable these can be.

For this, we model the probability distribution of the error, *i.e.*, the euclidean distance between ground truths y and forecasts $\hat{y}$, with an exponential distribution following [162]:

$$\|y - \hat{y}\|_2 \sim \mathrm{Exp}(\alpha), \tag{5.1}$$

where α is the distribution parameter to be selected. Its log-likelihood therefore writes

$$\ln p(\|y - \hat{y}\|_2) = \ln \alpha - \alpha \|y - \hat{y}\|_2 . \tag{5.2}$$

We then define the aleatoric uncertainty as $u := -\ln \alpha$, and set it as a learnable parameter for the model. When training the model with maximum likelihood estimation, the loss function $\mathcal{L}$ to minimize is then given by

$$\begin{aligned} \mathcal{L}(y, \hat{y}, u) &= -\ln p(\|y - \hat{y}\|_2) \\ &= e^{-u} \|y - \hat{y}\|_2 + u. \end{aligned} \tag{5.3}$$

We consider pose forecasting as a multi-task learning problem with task-dependant uncertainty, *i.e.*, independent of the input sequences x. There are several ways to define tasks in this manner, *e.g.*, by separating them based on time frames, joints, actions (if the datasets provide them), or any other combination of them. In the following, we consider dividing tasks based on time and joints [4]. In this case, for each future time frame t and joint j, the model predicts an uncertainty estimate u_t^j associated with its 3D joint forecasts $\hat{y}_t^j$. This formulation yields the corresponding loss function:

$$\mathcal{L}_{total}(y, \hat{y}, u) = \sum_{\substack{t=1...T \\ j=1...J}} e^{-u_t^j} \left\| y_t^j - \hat{y}_t^j \right\|_2 + u_t^j, \tag{5.4}$$

where T refers to the number of prediction frames and J is the number of joints.

Since the loss function (Equation (5.4)) weighs the error $\left\| y_t^j - \hat{y}_t^j \right\|_2$ based on the aleatoric uncertainty $e^{-u_t^j}$, it forces the model to focus its capacity to points with lower aleatoric uncertainty. In particular, we expect short time horizons to have lower uncertainty, and therefore to present better improvements than longer ones.

Unfortunately, learning all aleatoric uncertainty values u_t^j independently leads to an unstable training. To address this issue, we introduce uncertainty priors F, in order to inject knowledge about the aleatoric uncertainty behavior and stabilize the training. For this, we choose a family F of functions parameterized by a given number of parameters θ. Instead of learning all uncertainty values u_t^j independently, the model now only learns θ, which can be chosen to be of a smaller size so as to ease the training. With a learned θ^*, the uncertainty values u_t^j are obtained with the

[4] Extending the formulation to other task definitions should be straightforward.

function $F(\theta^*)$:

$$u_t^j = F(\theta^*)(j,t). \tag{5.5}$$

It is noticeable that this framework generalizes the previous case (without prior) by setting F to yield a separate parameter for each uncertainty value:

$$u_t^j = \mathrm{Id}(\theta^*)(j,t) = \theta_t^j. \tag{5.6}$$

Intuitively, the more parameters F has, the more scenarios it can represent, but at the cost of stability. We, therefore, compare several choices for F, with variable numbers of learnable parameters as different trade-offs between ease of learning and representation power. We select three functions that constrain the temporal evolution of aleatoric uncertainty, independently for each joint. We select functions with a logarithmic shape due to the observed exponential behavior in error evolution over time. The first one, Sig_3, is a sigmoid function used to ensure that uncertainty only increases with time, and has three parameters per joint to control this behavior:

$$u_t^j = \mathrm{Sig}_3(\theta)(j,t) = \frac{\theta_2^j}{1+e^{-\theta_0^j(t-\theta_1^j)}}. \tag{5.7}$$

Then we leverage Sig_5, which is a generalized version of the sigmoid function [198] with 5 parameters per joint:

$$u_t^j = \mathrm{Sig}_5(\theta)(j,t) = \theta_0^j + \frac{\theta_1^j}{1+ab+(1-a)c}, \tag{5.8}$$

where the terms a, b and c are defined by

$$\begin{aligned} a &= \frac{1}{1+e^{-\frac{2\,\theta_2^j\,\theta_4^j}{|\theta_2^j+\theta_4^j|}(\theta_3^j-t)}}, \\ b &= e^{\theta_2^j(\theta_3^j-t)}, \\ c &= e^{\theta_4^j(\theta_3^j-t)}. \end{aligned} \tag{5.9}$$

We also compare with a more generic polynomial function Poly_d of degree d, which has $d+1$ learnable parameters per joint and constrain the uncertainty less:

$$u_t^j = \mathrm{Poly}_d(\theta)(j,t) = \theta_0^j + \theta_1^j t + \theta_2^j t^2 + ... + \theta_d^j t^d. \tag{5.10}$$

5.4 Epistemic Uncertainty in Pose Forecasting

We introduced aleatoric uncertainty to address the uncertainty in the data in the previous section. We now address the epistemic uncertainty to capture the model's uncertainty due to the lack of

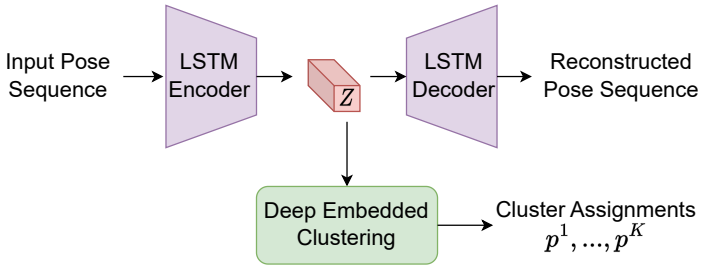


Figure 5.2: The motion is encoded into a well-clustered representation space Z by our LSTM encoder-decoder. The probabilities of the cluster assignments are provided by our deep embedded clustering on that space to estimate the epistemic uncertainty.

knowledge. We want to quantify the intuition that the models with predicted motions dissimilar to the training distribution in the latent representation are less reliable and, therefore, should be treated with caution. Notably, our aim in this section is not to improve accuracy but rather to measure uncertainties associated with pose forecasting models.

We improve upon existing literature of uncertainty quantification by introducing temporal modeling and clustering in epistemic uncertainty. Specifically, we employ an LSTM-based autoencoder (Figure 5.2) due to its proficient capability to encode spatio-temporal dependencies and learn potent latent representations. We then rely on clustering on that space as there are no predefined motion classes.

In the following sections, we first explain how to estimate the number of motion clusters K and train the deep embedded clustering. We then illustrate how to measure the epistemic uncertainty.

5.4.1 Determining the Number of Motion Clusters

Determining *K*, the number of clusters, is essential since it corresponds to the diversity of motions in the training dataset. An optimal *K*, therefore, captures the diversity in the training dataset while also reducing the time complexity of our subsequent algorithms.

We first train an LSTM auto-encoder (Figure 5.2) to learn low dimensional embeddings Z by minimizing the reconstruction loss L_{recons} over the training dataset. We then follow DED [199] which uses t-SNE [200] to reduce Z to a 2-dimensional feature vector z'. Subsequently, local density ρ_i and delta δ_i for each data point are calculated:

$$\begin{aligned} \rho_i &= \sum_j \chi(d_{ij} - d_c), \\ \delta_i &= \min_{j:\rho_j > \rho_i} d_{ij}, \end{aligned} \tag{5.11}$$

where $\chi(.) = 1$ if $. < 0$ else $\chi(.) = 0$, d_{ij} is the distance between z'_i and z'_j, and d_c is the cut-off distance. We then define $\gamma_i = \rho_i.\delta_i$ similar to [201], [202]. A larger γ_i corresponds to a greater

likelihood of being chosen as a cluster center; however, the number of clusters still remains a hyperparameter. We fully automate it by defining r_i as the gap between two γ_i and γ_{i+1} values (where $\gamma_{i+1} < \gamma_i$):

$$r_i = \frac{\gamma_i}{\gamma_{i+1}}, i \in [1, N-1]. \tag{5.12}$$

We set $K = argmax(r_i)$ since γ_K represents the largest shift in likelihood of a sample being a cluster itself.

5.4.2 Deep Embedded Clustering

Having identified the number of clusters, we now learn the optimal deep clustering of our embedding. We initialize the cluster centers $\{\mu^k\}_{k=1}^K$ using the K-means algorithm on the feature space. We then minimize the clustering loss $L_{cluster}$ as defined in DEC [203] jointly with the reconstruction loss in order to learn the latent representation as well as clustering. We incorporated the reconstruction loss into the loss function to act as a regularizer and prevent the collapse of the network parameters. The loss function is defined as:

$$L = L_{cluster} + \lambda L_{recons}, \tag{5.13}$$

where λ is the regularization coefficient. Finally, when the loss is converged, we fine-tune the trained network using the cross-entropy loss on the derived class labels in order to make clusters more compact.

5.4.3 Estimating Epistemic Uncertainty

Now, we estimate the epistemic uncertainty of a given forecasting model. Specifically, for each example, denote the probability of assignment to the kth cluster by p^k. The epistemic uncertainty is then calculated as follows:

$$EpU = \frac{1}{N} \sum_{i=1}^{N} \text{entropy}(p_i^1, \ldots, p_i^K), \tag{5.14}$$

where N is the size of the dataset. In other words, a model that does not generate outputs close to the motion clusters is considered uncertain.

5.5 Experiments

5.5.1 Datasets

Human3.6M [140] contains 3.6 million body poses. It comprises 15 complex action categories, each one performed by seven actors individually. The validation set is subject-11, the test set is subject-5, and all the remaining five subjects are training samples. The original 3D pose skeletons in the dataset consist of 32 joints. Similar to previous works, we have 10/50 observation frames, 25 forecast frames down-sampled to 25 fps, with the subset of 22 joints to represent the human pose. We train our models on all action classes at the same time.

AMASS (The Archive of Motion Capture as Surface Shapes) [166] unifies 18 motion capture datasets totaling 13,944 motion sequences from 460 subjects performing a variety of actions. We use 50 observation frames down-sampled to 25fps with 18 joints, similar to previous works.

3DPW (3D Poses in the Wild) [167] is the first dataset with accurate 3D poses in the wild. It contains 60 video sequences taken from a moving phone camera. Each pose is described as an 18-joint skeleton with 3D coordinates similar to AMASS dataset. We use the official instructions to obtain training, validation, and test sets.

5.5.2 Evaluation Metrics

We measure the accuracy in terms of MPJPE (Mean Per Joint Position Error) in millimeters (mm) per frame:

$$MPJPE = \frac{1}{J}\sum_{j=1}^{J}\left\|\hat{y}_t^j - y_t^j\right\|_2, \tag{5.15}$$

and report A-MPJPE as the average for all frames when needed. We also report EpU as defined in Equation (5.14).

5.5.3 Baselines

We apply our approach to several recent methods that are open-source [158]–[160] and compare with their performance without uncertainty. Note that we follow their own training setup in which some use 10 frames of observation [159], [160], [182] and the rest 50 frames of observation [158], [170], [172], [173]. We report the results obtained from the trained model of STARS* [204] as documented on their GitHub page. We also consider *Zero-Vel*, which outputs the last observed pose as the forecast for all future poses as a simple and competitive baseline.

Inspired by the common trend to treat sequences with Transformers, we have designed our own transformer-based architecture referred to as *ST-Trans*. We followed the best practices of

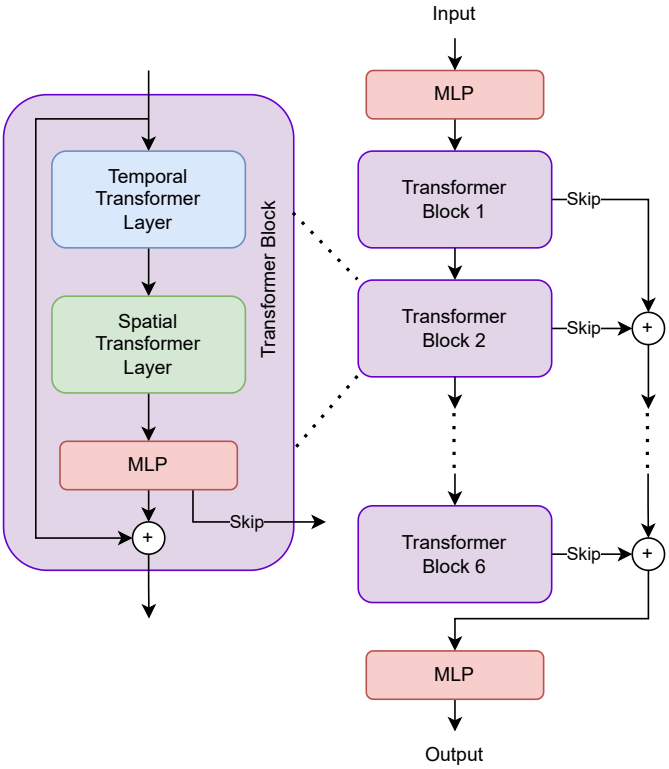


Figure 5.3: ST-Trans consists of two MLP layers and six Transformer Blocks with skip connections. Each Transformer Block contains two cascaded temporal and spatial transformers to capture the spatio-temporal features of data.

transformer design [205] and adapted their design elements to the task of pose forecasting. As depicted in Figure 5.3, it is composed of several identical residual layers, each layer consists of a spatial and a temporal transformer encoder to learn the spatio-temporal dynamics of data utilizing the attention mechanism.

5.5.4 Aleatoric Uncertainty

We first show the impact of aleatoric Uncertainty-Aware Loss (pUAL) with the prior Sig_5 to several models from the literature and our ST-Trans. Table 5.1 shows the overall results on Human3.6M [140]. To have a fair evaluation between all models, we adapt HRI [158] to predict 25 frames in one step (denoted as HRI*). We observe that all methods get better results when taking aleatoric uncertainty into account during learning, therefore confirming the need for aleatoric uncertainty estimation. It is noticeable that pUAL gives better improvements for shorter prediction horizons, *e.g.*, up to 25.4 % and 20.1 % for STS-GCN [159] at horizons of $80\,\mathrm{ms}$ and $160\,\mathrm{ms}$, which correspond to the less uncertain time frames, where pUAL focuses training more (smaller discount in the loss function, as seen in Equation (5.4)). At the same time, adding pUAL does not degrade the performances at longer horizons. Examples of predicted 3D pose sequences

Model	80 ms	160 ms	320 ms	400 ms	560 ms	720 ms	880 ms	1000 ms
Zero-Vel	23.8	44.4	76.1	88.2	107.4	121.6	131.6	136.6
Res. Sup. [172]	25.0	46.2	77.0	88.3	106.3	119.4	130.0	136.6
ConvSeq2Seq [170]	16.6	33.3	61.4	72.7	90.7	104.7	116.7	124.2
LTD-50-25 [173]	12.2	25.4	50.7	61.5	79.6	93.6	105.2	112.4
MSR-GCN [182]	12.0	25.2	50.4	61.4	80.0	93.9	105.5	112.9
STARS* [204]	12.0	24.6	49.5	60.5	78.6	92.6	104.3	111.9
STS-GCN [159]	17.7	33.9	56.3	67.5	85.1	99.4	109.9	117.0
STS-GCN + pUAL (ours)	13.2	27.1	54.7	66.2	84.5	97.9	109.3	115.7
gain	**25.4 %**	20.1 %	2.8 %	1.9 %	0.7 %	1.5 %	0.5 %	1.1 %
HRI* [158]	12.7	26.1	51.5	62.6	80.8	95.1	106.8	113.8
HRI* + pUAL (ours)	11.6	25.3	51.2	62.2	80.1	93.7	105.0	112.1
gain	**8.7 %**	3.1 %	0.6 %	0.6 %	0.9 %	1.5 %	1.7 %	1.5 %
PGBIG [160]	10.3	22.6	46.6	57.5	76.3	90.9	102.7	110.0
PGBIG + pUAL (ours)	9.6	21.7	46.0	57.1	75.9	90.3	102.1	109.5
gain	**6.8 %**	4.0 %	1.3 %	0.7 %	0.5 %	0.7 %	0.6 %	0.5 %
ST-Trans	13.0	27.0	52.6	63.2	80.3	93.6	104.7	111.6
ST-Trans + pUAL (ours)	10.4	23.4	48.4	59.2	77.0	90.7	101.9	109.3
gain	**20.0 %**	13.3 %	8.0 %	6.3 %	4.1 %	3.1 %	2.7 %	2.1 %

Table 5.1: Comparison of our method on Human3.6M in MPJPE (mm) at different prediction horizons. +pUAL refers to models where aleatoric uncertainty is modeled.

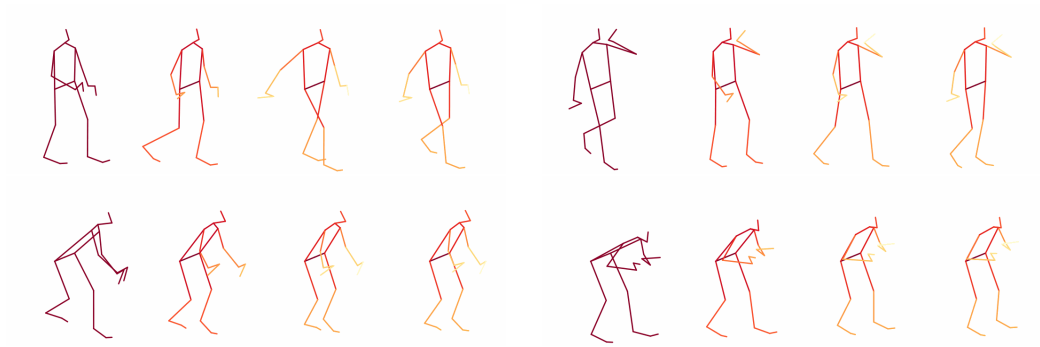


Figure 5.4: Qualitative results on Human3.6M different actions (walking, phoning, walkingdog and taking photo, respectively). Higher aleatoric uncertainty is shown with a lighter color. Uncertainty of any bone is considered as its outer joint's uncertainty assuming the hip is the body center. We observe that the estimated uncertainty increases over time, with joints farther away from the body center associated with higher uncertainties.

Model	AMASS				3DPW			
	160 ms	400 ms	720 ms	1000 ms	160 ms	400 ms	720 ms	1000 ms
Zero-Vel	56.4	111.7	135.1	119.4	41.8	79.9	100.5	101.3
ConvSeq2Seq [170]	36.9	67.6	87.0	93.5	32.9	58.8	77.0	87.8
LTD-10-25 [173]	20.7	45.3	65.7	75.2	23.2	46.6	65.8	75.5
STS-GCN [159]	20.7	43.1	59.2	68.7	20.8	40.3	55.0	62.4
STS-GCN + pUAL (ours)	20.4	42.4	59.1	68.1	20.5	40.0	54.8	62.2
HRI [158]	20.7	42.0	58.6	67.2	22.8	45.0	62.9	72.5
HRI + pUAL (ours)	19.9	41.4	58.1	66.5	22.2	44.6	62.4	72.2
ST-Trans	21.3	42.5	58.3	66.6	24.5	47.4	64.6	73.8
ST-Trans + pUAL (ours)	18.3	39.7	56.5	66.7	22.3	45.7	63.6	73.2

Table 5.2: Comparison of our proposed method on AMASS and 3DPW in MPJPE (mm) at different prediction horizons. +pUAL refers to models where aleatoric uncertainty is modeled. The models were only trained on AMASS.

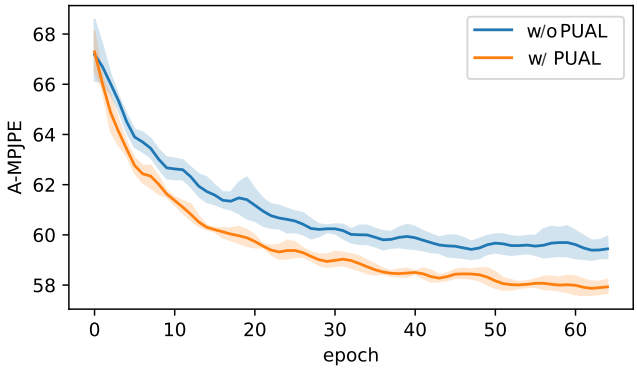


Figure 5.5: A-MPJPE and its standard deviation (stability) in training epochs for five trained models. The model with pUAL has a lower variance, meaning a more stable training.

using pUAL are depicted in Figure 5.4, and show that the estimated uncertainty increases over time, with joints farther away from the body center associated with higher uncertainties. Moreover, we report the performances of the models on AMASS and 3DPW datasets in Table 5.2. Again, we observe that modeling aleatoric uncertainty leads to more accurate predictions, especially at shorter horizons, with improvements up to 14.1 % on AMASS and up to 9.0 % on 3DPW for ST-Trans at a horizon of 160 ms.

We argue that modeling the aleatoric uncertainty leads to more stable training. In order to demonstrate this, we conduct five separate trainings of ST-Trans and present in Figure 5.5 the A-MPJPE values along with their respective standard deviations for each epoch, which we refer to as stability. The plot highlights that the model with pUAL is more stable across runs, as indicated by a lower variance. Moreover, we compute AP-MPJPE, which is the average pairwise distance of predicted motions in terms of MPJPE, and observe that it decreases from 24.2 mm to 20.3 mm when pUAL loss is added, showing again lower variance in the output of the model.

Uncertainty prior (tasks)	Number of parameters	stability	ST-Trans	HRI*	STS-GCN
None	–	0.643	111.6	113.8	117.0
Id (T, J)	$25 \cdot 22$	0.557	**109.3**	114.6	115.8
Poly_9 (T, J)	$10 \cdot 22$	0.505	110.3	114.7	118.1
Sig_5 (T, J)	$5 \cdot 22$	**0.496**	**109.3**	**112.1**	**115.7**
Sig_3 (T, J)	$3 \cdot 22$	0.537	110.3	113.1	115.9
Sig_5 (T)	5	0.505	109.7	112.4	115.9

Table 5.3: Comparison of different priors for aleatoric uncertainty in terms of MPJPE (mm) at $1\,\mathrm{s}$ on Human3.6M. The stability refers to the variance of the training and the lower the better.

So far, results have been reported using the Sig_5 uncertainty prior (Equation (5.8)) to model the time and joint (T, J) aleatoric uncertainty. In Table 5.3, we report the performances of other choices on Human3.6M, and compare against using a single prior Sig_5 for all joints (only time dependency T) and other priors Sig_3, Poly_9. The results show again that taking aleatoric uncertainty into account with pUAL is beneficial and that a good choice of uncertainty prior is important. In particular, Sig_5 performs better than using no prior for all models. Using a prior can lead to similar aleatoric uncertainty than the unconstrained case, but with fewer learnable parameters and better stability.

5.5.5 Epistemic Uncertainty

Evaluating the quality of epistemic uncertainty is difficult due to the unavailability of ground truth annotations, yet important. Our goal is to identify instances where pose forecasting is not reliable, essentially making this a binary classification problem. Selective classification is a widely used methodology to evaluate uncertainty quality, where a classifier has the option to refrain from classifying data points if its confidence level drops below a certain threshold [206]. In other words, if a pose forecasting model is trained on action A and evaluated on actions A and B, a reliable measure of epistemic uncertainty should effectively distinguish between these two sets of forecasts.

We assess the performance of our epistemic uncertainty estimation using selective classification, and measure how well actions "sitting" and "sitting down" can be separated from actions "walking" and "walking together", all from the test set of Human3.6M, based solely on the predicted uncertainty of the model. The forecasting model and clustering are trained on Human3.6M walking-related actions, and we anticipate low uncertainty values for those actions and high uncertainty values for sitting-related actions, *i.e.*, not encountered and significantly distinct actions. During the assessment, we compute uncertainty scores for both actions and measure the classification results for a range of thresholds. Similar to prior research [207], we utilize the AUROC metric, where a higher score is desirable and a value of 1 indicates that all walking-related data points possess lower uncertainty than all sitting-related data points. In Table 5.4, we present our

Method	AUROC	Latency	Trainings
Deep-Ensemble-3	0.87	6.28	3
Deep-Ensemble-5	0.90	10.43	5
MC-Dropout-5	0.90	9.57	1
MC-Dropout-10	0.92	18.98	1
Ours	**0.95**	**6.23**	1

Table 5.4: AUROC, inference latency (ms) and number of training runs for different epistemic uncertainty methods.

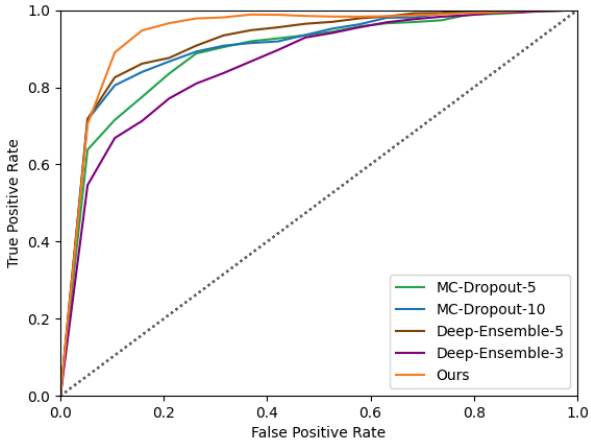


Figure 5.6: ROC curve for a model trained on walking-related actions and tested on both walking-related and sitting-related actions. The objective is to distinguish between these sets by utilizing uncertainty estimates.

findings and compare them to alternative approaches, where our proposed method demonstrates higher AUROC. The full ROC curve is in Figure 5.6. Note that our approach is model-agnostic in contrast to MC-Dropout. We present the results of further assessments of a broader range of actions in the Appendix C.2.

Another feature of our approach is computational efficiency, which is attributed to its ability to compute in a single forward path. This is in contrast to MC-Dropout and Ensemble methods. We provide a comparison of the average inference latency, measured in milliseconds, between our method and other approaches in Table 5.4. Our approach shows lower latency and only requires one training. Notably, the performance gap between our approach and other methods may increase when using more computationally expensive forecasting models.

We conducted another experiment in Table 5.5 to showcase the effectiveness of our metric in out-of-distribution (OOD) motions. We shuffled the frames' order (Frames shuffled) or joints (Joints shuffled) in each pose sequence of the test set to generate OOD data. The results demonstrate that our method identifies high uncertainties for Joints shuffled of all categories since they do

Action	Normal	Frames shuffled	Joints shuffled
Walking	0.26	1.35	2.15
Smoking	0.80	1.54	2.17
Posing	0.93	1.57	2.17
Directions	0.93	1.39	2.20
Greeting	0.81	1.50	2.17
Discussion	0.80	1.31	2.19
Walkingtogether	0.33	1.36	2.21
Eating	0.83	1.27	2.19
Phoning	0.82	1.56	2.20
Sitting	1.12	1.75	2.23
Waiting	0.82	1.57	2.15
Sittingdown	1.18	1.89	2.16
WalkingDog	0.95	1.53	2.20
TakingPhoto	1.02	1.47	2.17
Purchases	0.99	1.47	2.24
Average of all actions	**0.85**	**1.53**	**2.18**

Table 5.5: Comparison of EpU on different categories of Human3.6M. *Normal* refers to the original test set, *Frames shuffled* refers to the test set in which the frame orders in each sequence have been randomly shuffled, and *Joints shuffled* refers to randomly shuffled 3D joints in all frames.

	AMASS		3DPW	
Model	EpU	A-MPJPE	EpU	A-MPJPE
Zero-Vel	0.449	85.72	0.566	64.44
HRI [158]	0.351	43.76	0.463	43.62
STS-GCN [159]	0.332	45.49	0.455	42.60
ST-Trans + pUAL	0.336	35.86	0.439	40.02

Table 5.6: Comparison of different models in terms of A-MPJPE and EpU on AMASS and 3DPW datasets. The clustering and forecasting models were trained on AMASS.

not correspond to ID (in-distribution) poses. Furthermore, our approach yields high EpU values for almost all actions of Frames shuffled, compared to normal pose sequences, emphasizing the significance of frame order in generating an ID motion.

Additionally, we report the forecasting models' performances in Table 5.6 in terms of A-MPJPE, along with the epistemic uncertainties EpU associated with their predictions, on both the AMASS and 3DPW datasets. Note that the forecasting models and the clustering method were trained on the AMASS dataset. Higher uncertainties were recorded on 3DPW as an unseen dataset while prediction errors were lower. It underscores the reliability of our uncertainty quantification approach and suggests that relying solely on a model's prediction errors may not provide a comprehensive assessment.

5.6 Conclusions

In this chapter, we focused on modeling the uncertainty of human pose forecasting. We suggested a method for modeling aleatoric uncertainty of pose forecasting models that could make state-of-the-art models uncertainty-aware and improve their performances. We showed the effect of uncertainty priors to inject knowledge about the behavior of uncertainty. Moreover, we measured the epistemic uncertainty of pose forecasting models by clustering poses into motion clusters, which enables us to evaluate the trustworthiness of victim models. We made an open-source library of human pose forecasting. It incorporates various models, metrics, and supports multiple datasets, all aimed at fostering a unified and fair evaluation framework in this field. As future work, we hope that the findings and the library will pave the way to more uncertainty-aware pose forecasting models.

Moving forward, we extend our focus to situations that are even more challenging than those considered in this chapter. In the following chapter, we deal with the problem of pose forecasting under noisy observations.

6 Human Pose Forecasting in Noisy Observations

This chapter is based on the article:

Saeed Saadatnejad, Ali Rasekh, Mohammadreza Mofayezi, Yasamin Medghalchi, Sara Rajabzadeh, Taylor Mordan and Alexandre Alahi, *A generic diffusion-based approach for 3D human pose prediction in the wild*, IEEE International Conference on Robotics and Automation (ICRA), 2023

The code and a summary video of the work can be found on the project's webpage[1].

6.1 Introduction

As mentioned in the previous chapter, pose forecasting is a challenging task since it must combine spatial and temporal reasoning to output multiple plausible outcomes. Previous models have yielded satisfactory results [158], [160], yet they fail to produce acceptable outcomes in noisy settings. Minor offsets from detection methods or partial occlusions of body parts can drastically impact the prediction accuracy.

Denoising Diffusion Probabilistic Models (DDPMs) [208] are one type of generative models that can denoise input signals iteratively. Motivated by this property, we propose a diffusion model that explicitly handles noisy data input so that it not only predicts accurate and in-distribution poses, but can also be used in the wild.

As depicted in Figure 6.1, we construct a full sequence of observation and future frames where noise is placed in the missing observation elements and future poses. Our model denoises this sequence in several steps and produces the correct predictions. Naively predicting all future frames simultaneously results in inaccurate predictions in later frames. Hence, we propose a model comprised of two temporally-cascaded diffusion blocks. The first block predicts the short-term poses and repairs the noisy observations (if applicable), while the second block uses

[1] https://github.com/vita-epfl/DePOSit

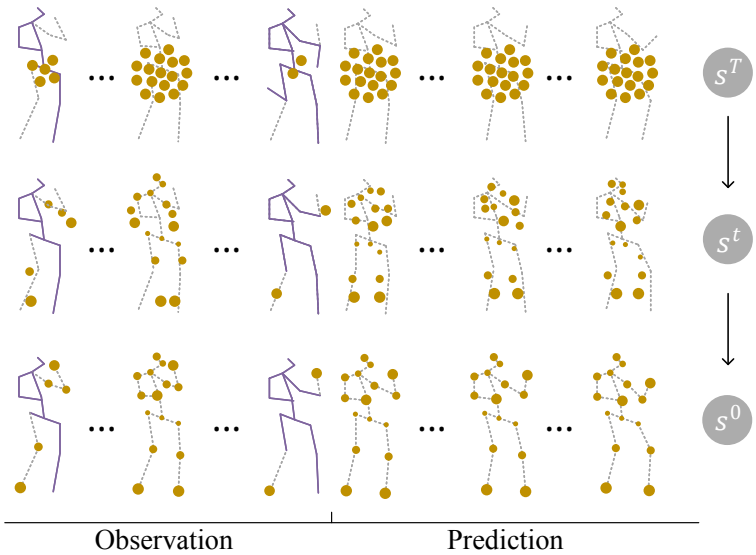


Figure 6.1: Our proposed conditional diffusion model denoises the input sequence s^T over T steps by simultaneously 1) predicting poses for the future frames and 2) repairing the noisy observations in the case of partial occlusion (first column), missing whole frame (second column), or inaccurate observations (third column). The large yellow circles depict the Gaussian noise we consider for unavailable joints, which gradually become smaller and fit into the correct locations.

the output from the former as a condition to predict the long-term poses.

We also leverage our model in a generic framework that can improve the performance of state-of-the-art prediction models in a black-box manner. To this end, we use our diffusion-based model as a pre-processing step to repair the observations providing pseudo-clean data for the prediction model to make more reliable predictions. Our model can then be used as a post-processing step to further refine these predictions.

To summarize, our contributions are three-fold:

- We frame the human pose forecasting task as a denoising problem.
- We propose a two-stage diffusion model outperforming the state-of-the-art in both clean and noisy observation settings.
- We introduce a generic framework that leverages our model through pre-processing (repairing the input) and post-processing (refinement), which can enhance any pose forecasting model.

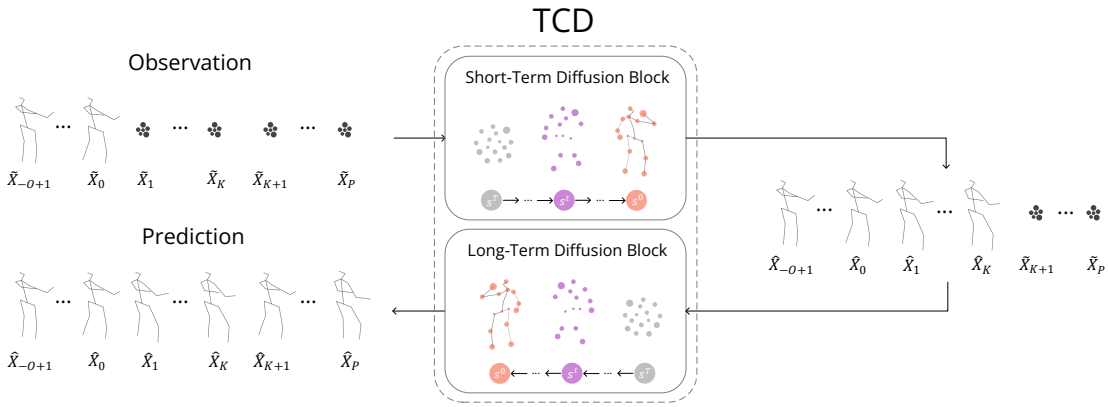


Figure 6.2: Overview of our Temporal Cascaded Diffusion (TCD). The short-term diffusion block (top) takes the observed sequence padded with random noise and predicts short-term human poses in K frames. The predicted sequence along with the observation padded with random noise is given to the long-term diffusion block (bottom) to predict for all P frames.

6.2 Related Work

6.2.1 Stochastic Human Pose Forecasting

As mentioned in Section 5.2.1, deterministic models [158], [160] offer satisfactory prediction accuracy, yet they lack the ability to generate diverse and multi-modal outputs compared to stochastic models [175]–[180], [204]. In this category, Variational AutoEncoders (VAEs) have been widely adopted due to their strength in representation learning [148], [175]–[177]. Generative models, particularly diffusion models, have been recently utilized to model data distributions with remarkable results in image synthesis [21], [209], image repainting [210] and text-to-image generation [8], [211]. Recently, they have been used for time-series imputation [205], i.e., filling in missing elements. However, it was not explored for human motion. To the best of our knowledge, we are the first to propose a diffusion model for human pose forecasting, which outperforms both stochastic and deterministic models.

Previous models perform poorly with partial noisy observations. A multi-task learning approach has been recently suggested in [212] to address this issue, by implicitly disregarding noise in the data. We provide detailed comparisons with [212], and show that explicitly denoising the input leads to a generalizable solution, and that our temporally-cascaded diffusion blocks better capture the spatio-temporal relationships in the poses. Furthermore, we present a generic framework that can be used to improve any existing state-of-the-art model in a black-box manner.

6.3 Method

In this section, we first describe the notations and conditional diffusion blocks, which are the fundamental elements of our model. We then present our model and finally introduce our generic

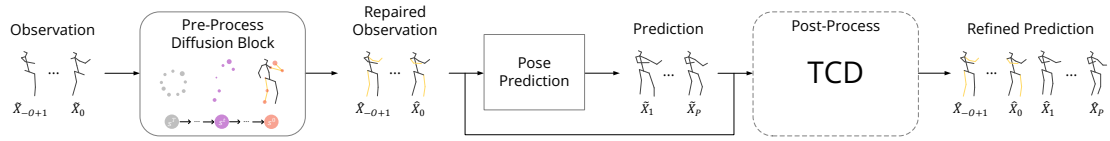


Figure 6.3: An illustration of the pre-processing and post-processing framework. The pre-process diffusion block denoises the noisy observation sequence. The repaired observation is then given to a frozen predictor. The output of the predictor model is passed to TCD to perform the post-processing step and refine its predictions.

framework.

6.3.1 Problem Definition and Notations

Let $X = [X_{-O+1}, X_{-O+2}, \ldots, X_0, X_1, \ldots, X_P] \in \mathbb{R}^{(O+P)\times J\times 3}$ be a clean complete normalized sequence of human poses with J joints in O frames of observation and P frames of future. Each joint consists of its 3D cartesian coordinates. The availability mask is a binary matrix $M \in \{0,1\}^{(O+P)\times J\times 3}$ where zero determines the parts of the sequence that are not observed due to occlusions or being from future timesteps. Note that the elements of M corresponding to P future frames are always zero. With this notation, the observed sequence $\tilde{X} = [\tilde{X}_{-O+1}, \tilde{X}_{-O+2}, \ldots, \tilde{X}_0, \tilde{X}_1, \ldots, \tilde{X}_P]$ is derived by applying the element-wise product of M into X and adding a Gaussian noise $\epsilon \sim \mathcal{N}(0,\ I)$ in non-masked area $\tilde{X} = M \odot X + (1-M)\epsilon$. The model predicts $\hat{X} = [\hat{X}_{-O+1}, \hat{X}_{-O+2}, \ldots, \hat{X}_0, \hat{X}_1, \ldots, \hat{X}_P]$ and the objective is lowering $|\hat{X} - X| \odot (1-M)$ given $\tilde{X}$.

6.3.2 Conditional Diffusion Blocks

A conditional diffusion block is an ST-Trans defined in the previous chapter, which contains multiple residual layers. Each layer consists of two consecutive transformers with the same input and output shapes. The first (temporal) transformer is responsible for modeling the temporal behavior of data. Its output is then fed to the second (spatial) transformer to attend to the human pose within each frame.

At training time, a Gaussian noise with zero mean and pre-defined variance is added to the input pose sequence s^0 to make a noisier version s^1. This process is repeated for T steps such that the output s^T will be close to a pure Gaussian noise in the non-masked area:

$$q(s^t|s^{t-1}) = M \odot s^{t-1} + (1-M) \odot \mathcal{N}(s^t; \sqrt{1-\beta^t} s^{t-1}, \beta^t \mathbf{I}), \tag{6.1}$$

where q denotes the forward process, and β^t is the variance of the noise in step t, determined using a scheduler. We use the cosine noise scheduler in our formulations, which was first introduced

in [213]:

$$\beta^t = 1 - \frac{f(t)}{f(t-1)}, \quad f(t) = \cos^2\left(\frac{t/T + c}{1+c} \cdot \frac{\pi}{2}\right), \tag{6.2}$$

where c is a small offset and is set to 0.008 empirically. The cosine noise scheduler provides a smoother decrease in input quality than other popular schedulers, such as quadratic and linear [213], enabling more accurate learning of step noise variances in our problem. The network learns to reverse the diffusion process and retrieve the clean sequence by predicting the cumulative noise that is added to s^t as described in DDPM [208].

At inference time, the model begins with an incomplete and noisy input sequence s^T, where Gaussian noise is put in the non-masked area and observed data in the masked area. Subsequently, the model iteratively predicts the poses $s^{T-1}, \ldots, s^0$ through an iterative process by subtracting the additive noise learned during training from the output of the preceding step, until a clean output approximating the ground truth is obtained.

6.3.3 Temporal Cascaded Diffusion (TCD)

We illustrate our main model, which consists of a short-term and a long-term diffusion blocks, in Figure 6.2. The short-term block takes $\tilde{X}$ as input and predicts the first K frames of the future $[\hat{X}_1 \ldots \hat{X}_K]$, along with the observation frames $[\hat{X}_{-O+1} \ldots \hat{X}_0]$. The long-term block is tasked with predicting the remaining frames of the future $[\hat{X}_{K+1} \ldots \hat{X}_P]$, utilizing both the observation and the output of the short-term block. Note that during training, both blocks are trained using ground-truth input; however, at inference time, the average of five samples of the short-term block is supplied to the long-term block.

Cascading two diffusion models improves overall and particularly long-term forecasting due to the division of the complex task. In other words, the short-term prediction block focuses on predicting a limited number of frames, and thanks to its accurate short-term predictions, the long-term prediction block acquires more data, thus allowing it to focus its capacity on longer horizons.

6.3.4 Pre-processing and Post-processing

Given a frozen pose forecasting model, we can enhance its performance through pre-processing by repairing its input sequence, and through post-processing by refining its outputs. This framework is illustrated in Figure 6.3.

Pre-processing Since most of the existing pose forecasting models are unable to handle noisy observations, we present a simpler version of our model that serves as a pre-processing step for denoising the observations only. This module takes the noisy observation sequence

	Human3.6M [140]				HumanEva-I [214]	
Model	A-MPJPE ↓	MPJPE ↓	MMADE ↓	MMFDE ↓	A-MPJPE ↓	MPJPE ↓
Pose-Knows [215]	461	560	522	569	269	296
MT-VAE [216]	457	595	716	883	345	403
HP-GAN [217]	858	867	847	858	772	749
BoM [218]	448	533	514	544	271	279
GMVAE [219]	461	555	524	566	305	345
DeLiGAN [220]	483	534	520	545	306	322
DSF [221]	493	592	550	599	273	290
DLow [175]	425	518	495	531	251	268
Motron [178]	375	488	–	–	–	–
Multi-Objective [179]	414	516	–	–	228	236
GSPS [180]	389	496	476	525	233	244
STARS [204]	358	445	442	471	217	241
TCD (ours)	**356**	**396**	**463**	**445**	**199**	**215**

Table 6.1: Comparison with stochastic models on Human3.6M Setting-A and HumanEva-I at the horizon of 2s.

$[\tilde{X}_{-O+1}, \tilde{X}_{-O+2}, \ldots, \tilde{X}_0]$ as input and outputs a repaired sequence $[\hat{X}_{-O+1}, \hat{X}_{-O+2}, \ldots, \hat{X}_0]$. The architecture of this model is similar to TCD, yet predicting within a single stage, with both the input and output sequences containing O frames. Our precise repair strategies allow any pose forecasting models trained on complete datasets to predict reasonable poses in noisy input conditions.

Post-processing Furthermore, we want to improve the prediction results of existing models. We feed the results of any black-box pose forecasting model $[\tilde{X}_1, \ldots, \tilde{X}_P]$ concatenated with repaired observation $[\hat{X}_{-O+1}, \hat{X}_{-O+2}, \ldots, \hat{X}_0]$ as the input to our TCD and retrain it to predict better. The initial prediction acts as the starting point that is gradually shifted toward the real distribution by our post-processing.

6.4 Experiments

6.4.1 Experimental Setup

Datasets

We evaluate the performance of all approaches on four widely-used pose forecasting datasets:

Human3.6M [140] is the largest benchmark dataset for human motion analysis, comprising 3.6 million body poses. For more details, refer to Section 5.5.1. The original pose skeletons in the dataset consist of 32 joints, but different subsets of joints have been used in previous works to represent human poses. To ensure a fair and comprehensive comparison, we consider three

different settings for the dataset as follows:

- **Setting-A**: 25 observation frames, 100 prediction frames at 50 frames per second (fps), with the subset of 17 joints to represent the human pose;
- **Setting-B**: 50 observation frames, 25 prediction frames down-sampled to 25 fps, with the subset of 22 joints to represent the human pose;
- **Setting-C**: 25 observation frames, 25 prediction frames down-sampled to 25 fps, with the subset of 17 joints to represent the human pose.

AMASS (Archive of Motion capture As Surface Shapes) [166] is a recently published human motion dataset. We use 50 observation frames down-sampled to 25 fps with 18 joints, as in previous studies.

3DPW (3D Poses in the Wild) [167] is the first dataset with accurate 3D poses in the wild. Each pose is described with an 18-joint skeleton, similar to the AMASS dataset. We use the official instructions to obtain training, validation, and test sets. For more details about these two datasets, refer to Section 5.5.1.

HumanEva-I [214] includes three subjects captured at 60 fps. Each person has 15 body joints. We remove the global translation and use the official train/test split of the dataset. The prediction horizon is 60 frames (1 second) given 15 observed frames (0.25 seconds), similar to [180].

Other Implementation Details

We train our models using the Adam optimizer [222], with a batch size of 32 and a learning rate of 0.001. The learning rate is decayed by a factor of 0.1 at 75% and 90% of the total epochs. Our model consists of 12 layers of residual blocks and 50 diffusion steps by default. In TCD, the length of short-term prediction K is set to 20% of the total prediction length P. Each transformer has 64 channels and 8 attention heads.

Evaluation Metrics

We measure the Displacement Error (DE), in millimeters (mm), over all joints in a frame. Then, we report A-MPJPE, which is the average DE across all prediction frames, and/or MPJPE, which is the DE in the final predicted frame. We also report the multi-modal versions of average DE (MMADE) and final DE (MMFDE), following [180]. We additionally report average DE for the missing joints of the observation frames in the repairing task (r-ADE).

Model	80ms	320ms	560ms	720ms	880ms	1000ms
Zero-Vel	23.8	76.0	107.4	121.6	131.6	136.6
Res. Sup. [172]	25.0	77.0	106.3	119.4	130.0	136.6
ConvSeq2Seq [170]	16.6	61.4	90.7	104.7	116.7	124.2
LTD-50-25 [173]	12.2	50.7	79.6	93.6	105.2	112.4
HRI [158]	10.4	47.1	77.3	91.8	104.1	112.1
PGBIG [160]	10.3	**46.6**	76.3	90.9	102.6	110.0
TCD (ours)	**9.9**	48.8	**73.7**	**84.0**	**94.3**	**103.3**
HRI [158] + TCD (ours)	10.3	47.3	72.9	83.8	94.0	102.9
PGBIG [160] + TCD (ours)	10.2	46.1	72.4	83.6	93.9	102.8

Table 6.2: Comparison with deterministic models on Human3.6M Setting-B in MPJPE (mm) at different horizons.

	AMASS [166]				3DPW [167]			
Model	560ms	720ms	880ms	1000ms	560ms	720ms	880ms	1000ms
Zero-Vel	130.1	135.0	127.2	119.4	93.8	100.4	102.0	101.2
convSeq2Seq [170]	79.0	87.0	91.5	93.5	69.4	77.0	83.6	87.8
LTD-10-25 [173]	57.2	65.7	71.3	75.2	57.9	65.8	71.5	75.5
HRI [158]	51.7	58.6	63.4	67.2	56.0	63.6	69.7	73.7
TCD (ours)	**49.8**	**54.5**	**60.1**	**66.7**	**55.4**	**61.6**	**67.9**	**73.4**

Table 6.3: Comparison with deterministic models on AMASS and 3DPW in MPJPE (mm) at different horizons.

6.4.2 Baselines

We compare our model with several recent methods, including stochastic [175], [178]–[180], [204] and deterministic approaches [158]–[160], [170], [172]–[174] when possible. Note that some methods are not open-source and have different settings than ours. We also include *Zero-Vel* as a competitive baseline.

6.4.3 Comparisons with the State of the Art

We separate our experiments into three different settings: we first compare to other stochastic approaches, then to deterministic ones, and finally evaluate on noisy scenarios, with missing or noisy observation data.

Comparisons with Stochastic Approaches

We evaluate our model on two datasets, Human3.6M [140] Setting-A and HumanEva-I [214], and compare it with other stochastic approaches in Table 6.1. Each model is sampled 50 times

given each observation sequence. TCD (ours) clearly outperforms all previous works in terms of accuracy of the best sample (as measured by A-MPJPE and MPJPE) and multiple samples (as measured by MMADE and MMFDE).

Comparisons with Deterministic Approaches

We then compare our model to deterministic approaches on Human3.6M [140] Setting-B, tabulated in Table 6.2. To compare with deterministic models, our model is sampled five times, and the best sample is considered. Our proposed model surpassed previous works in the short-term and with a marked margin in the long-term, thanks to our two-stage prediction strategy. The detailed results of our model's performance on all categories of Human3.6M, along with comparisons with models that are not reported in standard settings, can be found in the appendix. We have also included the results of two previous state-of-the-art models that have been post-processed by our generic framework at the bottom of Table 6.2. Note that as the input data is complete, we only add post-processing (TCD) to their outputs. The improvements from our framework are non-negligible and can even beat our original model. Our two-stage prediction reveals a more pronounced benefit for longer horizons, which suggests that starting with a better initial guess can better shift the pose sequence toward the real distribution.

Substantial long-term improvement can be observed in AMASS [166] and 3DPW [167] as well. Similar to previous works, we train our model on AMASS and measure the MPJPE on both datasets. The comparison with models reporting in this setting is in Table 6.3. Note that for faster training, $K = 0$ was considered in this experiment.

Qualitative results on Human3.6M are shown in Figure 6.4. Predictions from our model are displayed along with predictions from several baselines and are superimposed on the ground-truth poses for direct comparison. Our model has successfully learned the data distribution, resulting in accurate and realistic poses; for instance, the hand movement is natural when the feet move while HRI has fixed hands and PGBIG has a momentum that avoids large hand movements. Moreover, post-processing can be used to further refine the predicted pose and shift it toward the ground truth.

Comparisons on Noisy Observation Data

We now examine the performance of models in the realistic scenario of noisy observations, since occlusions and noise are commonly seen in practice. To simulate occlusions, we remove 40% of the left arm and right leg from the observations of Human3.6M Setting-B, both during training and evaluation. The results in the top half of Table 6.4 show that the state-of-the-art models perform inadequately when the observation is noisy, whereas our model achieves results close to those of the clean input observation. Our pre-processing module repairs the observation sequences before feeding to the state-of-the-art models and Zero-Vel, resulting in significant improvements in forecasting performance. MT-GCN [212] was designed to provide accurate predictions in

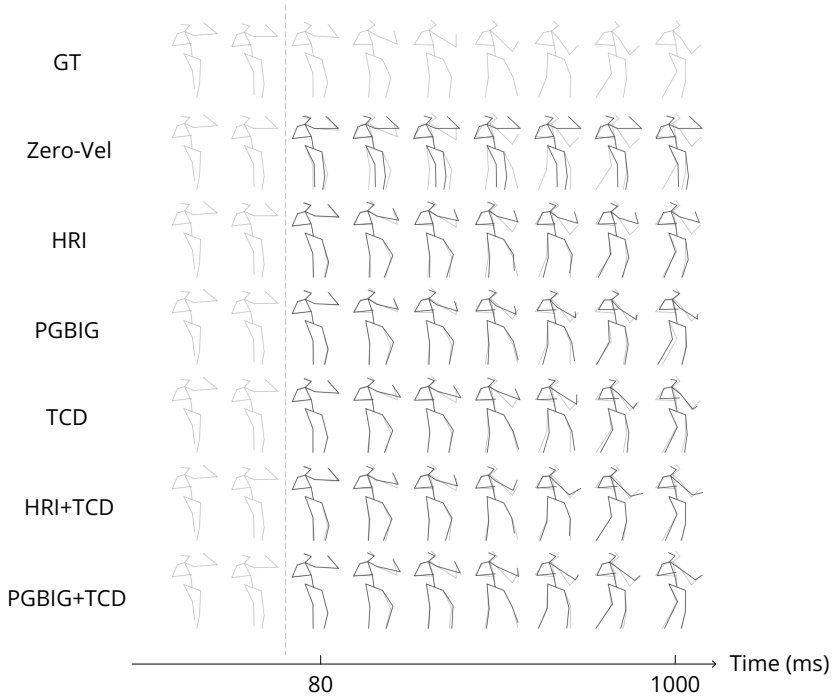


Figure 6.4: Qualitative results on Human3.6M Setting-B. The left part of each row shows the input observation, while the right part displays the predicted poses superimposed on the ground truth.

Model	80ms	320ms	560ms	720ms	880ms	1000ms
Zero-Vel	84.9	138.2	169.9	184.2	193.7	198.2
HRI [158]	65.2	104.5	130.0	141.6	151.1	157.1
PGBIG [160]	67.0	107.1	132.1	143.5	152.9	158.8
TCD (ours)	**11.2**	**51.3**	**75.4**	**85.4**	**95.4**	**104.5**
Pre(ours) + Zero-Vel	24.1	76.3	107.6	121.7	131.7	136.7
Pre(ours) + HRI [158]	11.4	48.6	78.3	92.7	105.0	112.8
Pre(ours) + PGBIG [160]	11.1	**47.9**	77.2	91.7	103.5	110.8
Pre(ours) + TCD (ours)	**10.8**	49.9	**74.4**	**84.9**	**95.1**	**104.2**

Table 6.4: Comparison on noisy observation data and pre-processed observation data (Pre(ours)+) on Human3.6M Setting-B in MPJPE (mm) at different horizons.

incomplete observations. We compared our model to it and some other prior models and present the results on Human3.6M Setting-C in the first column of Table 6.5. Our model achieved a remarkable improvement of 33.2mm in MPJPE at 1s horizon (30% improvement) over MT-GCN. It should be noted that the models in the upper part of the table received repaired sequences using MT-GCN's own preprocessing, while the rest received noisy sequences.

We analyzed the performance of our model in several occlusion patterns masks M that are applied

Model	Random Leg, Arm Occlusions	Structured Joint Occlusions	Missing Frames	Gaussian Noise $\sigma = 25$	Gaussian Noise $\sigma = 50$
R+TrajGCN [173]	121.1	131.5	–	127.1	135.0
R+LDRGCN [223]	118.7	127.1	–	126.4	133.6
R+DMGCN [224]	117.6	126.5	–	124.4	132.7
R+STMIGAN [225]	129.5	128.2	–	–	–
MT-GCN [212]	110.7	114.5	122.0	114.3	119.7
TCD (ours)	**77.5**	**77.2**	**80.5**	**81.9**	**84.9**

Table 6.5: Comparison on noisy observation data on Human3.6M Setting-C in MPJPE (mm) at the horizon of 1s. The upper part of the table contains models that received repaired sequences (R+), while the lower part contains models that received noisy sequences.

	Train and Test Missing Ratio			
Model	10%	20%	30%	40%
MT-GCN [212]	109.4 / 8.6	110.5 / 13.7	112.3 / 18.7	114.4 / 24.5
TCD (ours)	**77.1 / 2.2**	**77.2 / 2.3**	**77.6 / 2.6**	**79.1 / 2.9**

Table 6.6: Results of motion prediction and sequence repairing on Human3.6M Setting-C with varying amounts of randomly occluded joints in input data in MPJPE (mm) at the horizon of 1s / r-ADE (mm) of missing elements.

to input data:

- Random Leg, Arm Occlusions: leg and arm joints are randomly occluded with a probability of 40%;
- Structured Joint Occlusions: 40% of the right leg joints for consecutive frames are missing;
- Missing Frames: 20% of the consecutive frames are missing;
- Gaussian Noise: Gaussian noise with a standard deviation of $\sigma = 25$ or $\sigma = 50$ is added to the coordinates of the joints, and 50% of the leg joints are randomly occluded.

The results of training and evaluating our model on these observation patterns, in MPJPE at the prediction horizon of 1 second on Human3.6M Setting-C, are presented in Table 6.5. Our model outperformed previous works in different patterns of occlusions and noises in input that can occur in the real world. Furthermore, we observed that missing 5 consecutive frames is more challenging than missing a part of the body in 10 consecutive frames, as the network can recover the latter with spatial information.

To have a thorough comparison with MT-GCN, we trained four models by varying the percentage of joints randomly removed from the pose observation sequence. The performance of sequence

		Test Missing Ratio			
		0%	20%	50%	90%
Train Missing Ratio	20%	**76.8** / –	**77.2** / 2.3	79.9 / 5.2	159.3 / 114.7
	50%	78.7 / –	77.8 / **2.1**	**78.7** / **3.6**	105.2 / 51.9
	90%	82.3 / –	82.3 / 3.1	82.9 / 4.3	**89.5** / **21.8**

Table 6.7: Results of motion prediction and sequence repairing on Human3.6M [140] Setting-C with varying proportions of randomly occluded joints between training and testing in MPJPE (mm) at a horizon of 1s / r-ADE (mm) of missing elements.

repairing (r-ADE of the occluded observation sequence) and motion prediction (MPJPE at 1-second horizon) is presented in Table 6.6. Our model exhibited a negligible error of 2.9mm in repairing with up to 40% of all joints missing, whereas MT-GCN exhibited an error of 24.5mm. Indeed, our model achieved more than 31% lower MPJPE compared to MT-GCN in forecasting.

Moreover, in inference time, our model performs well even if it has not observed that occlusion pattern in training time. We show this in Table 6.7, where we train our model with varying percentage of random occluded joints and evaluate it with different percentages of random occlusion. We observe that less noise at test time than training naturally shows higher prediction and repairing performances, and more noise at test time than training weakens. On the other hand, learning on highly occluded observation data leads to a better generalization when testing with a similarly high level of occlusion but to a slight decrease in MPJPE when testing with low levels of occlusion. Our model is able to predict with 90% of missing input without a considerable degradation in performance. This can be a great benefit in real-world applications with imperfect data.

6.4.4 Ablations Studies

Here, we investigate different design choices of the network and report A-MPJPE on Human3.6M [140] Setting-B. For faster training, only a fifth of the dataset was utilized in this section. The full model yielded an A-MPJPE of 63.3mm. When predicting in one stage, without any subdivisions, the A-MPJPE increased to 65.5mm due to erroneous predictions in longer time frames. Conversely, when predicting in three stages, i.e., 20%, 20%, and 60%, the performance dropped to 66.9mm, as cascading multiple stochastic processes leads to either random outcomes or a lack of diversity. This illustrates the efficacy of two-stage prediction. Another important factor is the length of short-term prediction. In our experiments, a prediction of $P = 25$ frames was made with $K = 5$. A lower $K = 2$ reduced the benefits of two-stage prediction (A-MPJPE of 65.1mm). On the other hand, a higher $K = 10$ made short-term prediction more difficult, leading to an increased A-MPJPE of 66.6mm.

We tested a quadratic scheduler instead of our cosine scheduler and it increased A-MPJPE by 1mm. Our full model employed 12 residual layers in its diffusion blocks; however, decreasing

this number to 4 resulted in a decrease in performance by 3mm. We refrained from utilizing more than 12 residual layers due to the considerable negative influence on the sampling time. Moreover, we conducted several experiments on the architecture of the transformers and found that spatial transformer and time transformer both facilitated the learning of spatio-temporal features of the pose sequence. Eliminating either of these resulted in an A-MPJPE of 74.5mm and 261.1mm, respectively.

6.5 Conclusions

In this chapter, we proposed a denoising diffusion model for human pose forecasting suitable for noisy input observations occurring in the wild. Our model predicted future poses in two stages (short-term and long-term) to better capture human motion dynamics, achieved superior performance compared to the state-of-the-art on four datasets, including both clean and noisy input settings. We then leveraged it to create a generic framework that is easily applicable to any existing predictor in a black box manner in two steps: pre-processing to repair the observations and post-processing to refine the predicted poses. We have applied it to several previous predictors and enhanced their predictions. The high computational complexity of diffusion models is a well-known challenge, and future studies may explore ways to accelerate the model's performance without sacrificing accuracy.

Having tackled motion forecasting at both coarse and fine levels under varying conditions, we turn our gaze in the next chapter toward image synthesis. This interconnected concept can serve as a practical platform for testing our forecasting models within simulated scenarios.

7 Image Synthesis for Simulation

This chapter is based on the article:

Saeed Saadatnejad, Siyuan Li, Taylor Mordan, and Alexandre Alahi, *A Shared Representation for Photorealistic Driving Simulators*, IEEE Transactions on Intelligent Transportation Systems (T-ITS), 2021

The code and a summary video of the work can be found on the project's webpage[1].

7.1 Introduction

Safety is the primary concern when developing autonomous vehicles (AVs). For example, a wrong action in an unexpected situation can lead to a collision with a pedestrian, which is not negligible [148]. Yet, strictly evaluating AVs in the real world is not a realistic nor a safe option. Some argue that an AV should be tested millions of miles in challenging situations to demonstrate its performance [226]. Besides its extensive required time and costs, it is impossible to cover all rare cases. Simulations can play a significant role in overcoming these issues [227]. By synthesizing images, we are able to not only evaluate the performance of AVs but also improve the performance of current deep networks leveraging the abundant amount of data [228]–[232].

Researchers have investigated two paradigms: model-based and data-driven simulators. The former is based on physics laws and computer graphics, such as Carla [233]. It needs high-fidelity environmental models to create indistinguishable images, which is highly expensive. The latter learns to effectively generate the images from examples [234]–[236]. In this work, we tackle the semantically-driven image synthesis task: given a semantic mask (*e.g.*, human body poses, or scene segmentation masks), we aim to generate a realistic image with the same semantics.

Photorealistic image synthesis is a notoriously difficult task due to a high dimensional output space and an ill-posed objective. It is commonly done with conditional Generative Adversarial

[1] https://github.com/vita-epfl/SemDisc

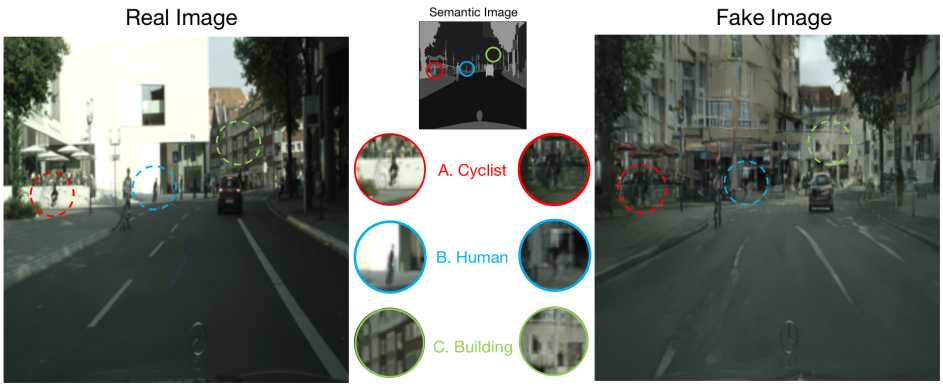


Figure 7.1: Given the appropriate semantic map, the network is supposed to synthesize a realistic image with the desired semantic. Although a fake image may look realistic from a global view, two problems remain: some semantics are not followed (A and B) and fine-grained details reveal the fake one (C).

Networks (cGANs) [234], [237]–[239]. However, state-of-the-art approaches cannot always provide enough supervision to the generator. As a solution, some provide a structured semantic description as another input to the discriminator. The discriminator of the cGAN is in charge of classifying the whole image as real or synthetic conditioned on the specified semantic input, hoping to learn the joint distribution of (image, segmentation). Yet, learning to make generated images realistic leads to not perfectly following the semantic content, especially for some small or rare objects, as shown in Figure 7.1. Another solution to attain high fidelity images is adding conditional matching losses in the pixel space. This is too strict as only the high-level description needs to be followed. Indeed, the discriminator is bypassed, and the generator is directly supervised by the content reconstruction. Finally, there also exists another problem in the main adversarial task. The discriminator gives the same weight to all image regions and does not learn a specialized network for the texture of a specific semantic class. For instance, what makes a car real might differ from what makes a road real.

We argue that the task of the discriminator, classifying a real/fake image, is closely related to having the capacity to understand its content *e.g.*, recognizing semantic, and compressing it. Hence, we ask the discriminator to perform three tasks: (1) image segmentation, to verify the loyalty of the generated image and the requested label, (2) reconstruction task, aiming at the conceptual understanding of the semantics, and (3) coarse-to-fine grained adversarial task trying to distinguish between fake and real in a class-specific manner. Since all these tasks share some useful information in the pixel-domain, we propose to learn them within the same representation as the adversarial supervision. We learn a shared latent representation that encodes enough information to jointly do semantic segmentation, content reconstruction, along with a coarse-to-fine-grained assessment. This leads to a more semantic-consistent output, more stable training, and more details in the images.

Our main contribution is learning a shared representation to provide correct supervisions for the generator. This is performed by a new architecture for the discriminator called SemDisc, in a multi-task learning approach, which is shown in Figure 7.2. The discriminator consists of three heads: the first head (semantics) forces the generator to follow the semantics explicitly, the second head (reconstruction) reconstructs the image back, acting as a regularizer to the training process, and the third head (coarse-to-fine adversarial) modifies the loss function to maintain coarse-to-fine grained details in the generated image. Finally, we introduce a trick to stabilize the training process.

The improvements we present in this chapter are generic and simple enough that any architecture of cGAN could benefit from a conversion from a regular discriminator to a structured semantic one. Interestingly, as only the discriminator is modified, it should be independent of the particular generator architecture used and should also be complementary to any approach based on generator enhancement, *e.g.*, [239], [234], [238].

Since the discriminator is used during training only, it is noticeable that all the changes we apply do not bring any run-time overhead, both in forward time and memory footprint. All effects happen through better learning of the model, thanks to the shared representation learning, compared to modifying the generator network, *e.g.*, by adding additional capacity to it that might impact the forward pass.

Finally, to show how our approach can influence the training of AVs, we share an in-depth analysis of our model on three image generation datasets that are related to transportation. This covers car-view image synthesis and building image synthesis from segmentation maps and human image synthesis from human poses (keypoints).

7.2 Related Work

7.2.1 Image Generation

While newer methods have demonstrated remarkable outcomes through the use of neural radiance fields [240], this research specifically focuses on image synthesis using GANs [241] and VAEs [242] due to their operational efficiency at the time of the study. GANs use two separate networks, a generator, and a discriminator that jointly optimize exclusive objectives. In this process, the generator learns to generate more realistic images, and the discriminator learns to distinguish between real and fake images more accurately. VAEs are another type of generative models that rely on probabilistic graphical models. Although they have been shown to disentangle features, the generated images are usually not as realistic as those from GANs [243] so in the remainder of this chapter, we mainly consider GANs.

Several methods have modified the design of the generator of GANs to get better results. Using mapping networks, adaptive instance normalization (AdaIN) [244] and spatially adaptive normal-

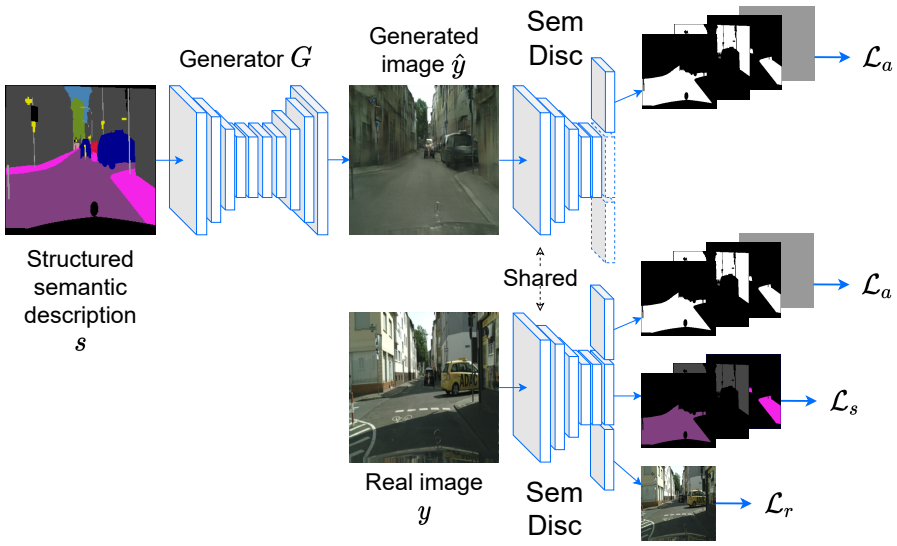


Figure 7.2: Conditional GAN training with semantic guiding. SemDisc has three heads. The first head provides several maps gated by semantic masks corresponding to the structured high-level description of the target output to focus learning on relevant areas. In the second head, the semantic is also leveraged to compute a semantic loss, matching the given constraint in a suitable space rather than a pixel one. The reconstruction head is supposed to reconstruct the image back, matching the texture in the pixel domain. The first head is trained on real and generated images, but other heads are trained only on real images.

ization (SPADE) [238] are among successful ideas in improving its architecture. These kinds of improvements have recently led to stunning results in the generation of natural images [245] or human faces [244], [246]. Moreover, it has been shown that these realistic generated images could be used as data augmentation in other tasks to improve accuracy, *e.g.*, in person re-identification [231], [230], semantic segmentation [247], [248] and even inspection of defect railway fasteners [249].

7.2.2 Conditional Image Generation

cGANs generate images from other images or high-level descriptions of the desired outputs. Applications can be various, as exemplified by pix2pix [237], which applies the image-to-image translation approach to a wide range of computer vision problems. More recently, realistic results have been obtained in the generation of city scenes using semantic maps [234], [238], [250], and even talking head videos from few examples [251].

To improve the quality of generated images, some added an auxiliary classification task [252], some tried to find and modify the regions of interest by means of attention maps [253], [254] and recently, some used pre-trained segmentation networks [255]. However, in [256], they showed that merely adding the segmentation loss (pixel-wise cross-entropy loss) leads to unstable training

with many artifacts. Indeed, they defined a baseline with an additional term in the loss function that when the synthesized image is given as input to a pretrained semantic segmentation network, it should produce a label map close to the input semantics.

7.2.3 Conditional Human Image Generation

In spite of realistic results in face image generation, human image synthesis is far from looking real since images need fine details of all body parts for a synthesized image to be considered as real. The problem becomes harder in conditional human image synthesis, where the model has to preserve the identity and texture of the conditioned image. One major issue is large body deformations caused by people's movements or changes in camera viewpoint. Several ideas have been developed. [257] added a pose discriminator, [239] introduced deformable skip connections in its generator and used a nearest neighbour loss, [258], [259] disentangled foreground people from background to transform them into the new pose while trying to have a background close to the source image. [260] learned a latent canonical view of a pedestrian in order to generate in any pose. [261] designed a soft-gated Warping GAN to address the problem of large geometric transformations in human image synthesis. [262], [263] trained a personalized model for each person, and [264] leveraged a few-shot learning approach needing few images from a new person to refine the network at test time.

7.2.4 Discriminator in Image Generation

The architecture of the discriminator plays a role in the quality of generated images through the learning of the generator. Patch-wise discriminators (PatchGANs) have outperformed global ones with full-image receptive fields for both neural style transfer [265] and conditional image generation [237]. Although the discriminator is often discarded after training, some methods leverage the information it learns. [234] yields high-quality images by having multiple discriminators at different resolutions, and [262] uses two separate networks for synthesizing full-body and face. [266] improves the quality of generated images and prevents mode collapse by leveraging the information stored in the discriminator and reshaping the loss function of GAN during image synthesis.

[267] also uses semantics to guide the discriminator but in a setup of image translation between domains (real $\Leftrightarrow$ virtual). Thus, they need images as input while we need semantics. Their approach is restricted to cases when the segmentation label maps cover the whole image. However, ours, by modifying the masking process and adding the coarse layer (responsible for making the whole image realistic), can work in all non-complete semantic maps such as human image synthesis. Moreover, we provide a multi-task learning approach with an added semantic matching head.

Recently, some others tried to leverage a U-net architecture in their discriminator. [268] provided detailed per-pixel feedback to the generator while maintaining the global context in an uncon-

ditional setting (without semantics). [269] modified the discriminator and defined a semantic alignment score map derived by multiplying activations of different layers of the discriminator with the ground truth label map. This strict constraint, which acts as a regularizer, could slightly improve the scores.

7.3 Method

We propose a multi-task learning approach to address conditional Generative Adversarial Network (cGAN) training for general-purpose image synthesis. We include structured semantic information to guide learning in order to focus more on meaningful regions of images. We build on successful cGAN models [238], [239] and propose to add the following appropriate supervisions: (i) biasing the discriminator toward semantic features, (ii) training a semantic matching, and (iii) adding a novel reconstruction loss which will subsequently influence the learning of the image generator network.

7.3.1 Overview of the Approach

Our model is composed of a main network G generating an image $\hat{y} = G(s)$ from a structured semantic description $s = (s_1, \ldots, s_K)$ over K feature maps (*e.g.*, class masks or heatmaps of keypoints) of the desired output, as depicted in Figure 7.2. During learning, examples consist of pairs (y, s) of real images y and their corresponding semantic descriptions s. After training, the distribution of generated images $\hat{y} = G(s)$ is expected to be similar to the distribution of y as they should share the same underlying semantic structures s. However, it is not easy to handcraft a loss function to assess the quality of the outputs $\hat{y}$ of G. For this, a discriminator network D is concurrently trained with it to act as a proxy loss, both networks competing to optimize exclusive loss functions in an adversarial minimax game [241].

As illustrated in Figure 7.2, our approach is different from common conditional GAN discriminators in which the image and the semantic map are concatenated. Indeed, the discriminator D takes as input an image x, and its semantic description s is applied in the loss function. The image x can either be generated by G (in which case $x = \hat{y}$) or be a real image ($x = y$), and D is trained to identify this, through minimization of an adversarial loss $\mathcal{L}_a$, a semantic matching loss $\mathcal{L}_s$ and a new reconstruction loss $\mathcal{L}_r$. At the same time, the generator G learns to generate images that both are realistic and match the input constraints.

In order to generate realistic images, the generator G learns to fool the discriminator D by maximizing its loss $\mathcal{L}_a$. Usually, the training of cGANs does not leverage all the semantic content of the description s. We suggest that properly incorporating structured semantics into the training should result in generated images with better details around these semantic features. For this, we modify the discriminator network D and its associated loss function $\mathcal{L}_a$, which will impact the training of the generator G, as detailed in Section 7.3.2.

The second objective to be optimized by the generator is having images matching their semantic descriptions. It is usually achieved by training G with the guidance of D. However, it only uses the description s as input to D to check whether the image matches it. To solve this issue, we split the task of D. The new head explicitly minimizes the semantic loss $\mathcal{L}_s$, described in Section 7.3.3. Moreover, to regularize the training, we define a novel reconstruction loss $\mathcal{L}_r$, described in Section 7.3.4.

The complete loss function $\mathcal{L}_G$ to be minimized by the generator network G is therefore

$$\mathcal{L}_G = -\mathcal{L}_a + \lambda_s \mathcal{L}_s + \lambda_r \mathcal{L}_r, \tag{7.1}$$

where λ_s and λ_r are weighting coefficients between those loss terms. For the discriminator, it is similar with increasing the adversarial loss

$$\mathcal{L}_D = \mathcal{L}_a + \lambda_s \mathcal{L}_s + \lambda_r \mathcal{L}_r. \tag{7.2}$$

Note that in our approach, D is composed of three heads D_a, D_s, D_r which share all layers except the last convolution layer, which will be described later.

7.3.2 Coarse-to-fine Adversarial Head

Our discriminator architecture is based on PatchGAN's one [265], whose output consists of a feature map where the score at each location indicates whether the corresponding input image patch is real or generated. PatchGAN discriminator D_{patch} is trained with a classification cross-entropy loss function $\mathcal{L}_{D_{patch}}$. We removed the semantics from its input which leads to

$$\begin{aligned}\mathcal{L}_{D_{patch}}(\hat{y}, y) =& \mathbb{E}_y \left[-\log\left(D_{patch}(y)\right)\right] \\ &+ \mathbb{E}_{\hat{y}} \left[-\log\left(1 - D_{patch}(\hat{y})\right)\right].\end{aligned} \tag{7.3}$$

However, instead of having a single output map globally classifying images, we here use multiple ones and force each of them to focus on a different semantic feature described by s.

Specifically, as illustrated in Figure 7.2, for a structure s with K channels, the coarse-to-fine adversarial head of the discriminator D_a outputs $K+1$ maps $D_a(x) = (D_{a_0}, D_{a_1}, \ldots, D_{a_K})$. The first map, D_{a_0}, handles the whole foreground objects described by the full tensor s at a coarse, global scale. Then, each map D_{a_k} of the remaining K ones corresponds to a given localized semantic feature s_k, in order to model fine-grained details associated with this feature.

To guide the learning of the various fine-grained prediction heads toward their corresponding semantic regions, semantic masks $M_k(s_k)$ are designed from the features s_k to indicate their locations within images. Note that the exact way semantic masks M_k are obtained from the features s_k depends on the type of their structures and is described in Section 7.4 for each dataset separately. The classification loss $\mathcal{L}_{D,k}$ used to train the branch k is then element-wise multiplied

with its associated mask M_k to select spatial areas that are relevant for the semantic feature attended to. Thus, backpropagation happens on the selected elements and their surroundings only, so that other regions of images not related to this feature do not affect the training. By explicitly attending to different semantic areas, it should be easier for the discriminator to focus on local details not easily captured by a global view on the image so that the generator learns to refine them. Regarding the coarse scale, the mask $M_0(s)$ is defined as covering the whole image and used in the same way. The complete loss function $\mathcal{L}_a$ for this head of the discriminator D_a is then the weighted sum, with equal weight given to the coarse loss than to all other fine-grained ones as these should only refine the first one,

$$\mathcal{L}_a = \mathcal{L}_{a,0} + \sum_{k=1}^{K} \frac{1}{K} \mathcal{L}_{a,k}, \tag{7.4}$$

where each term $\mathcal{L}_{a,k}$ is defined by the masked[2] version of the PatchGAN loss function from Equation (7.3):

$$\begin{aligned} \mathcal{L}_{a,k} = &\ \mathbb{E}_{y,s} \left[-\log \left(D_{a,k}(y) \right) \odot M_k(s_k) \right] \\ &+ \mathbb{E}_{\hat{y},s} \left[-\log \left(1 - D_{a,k}(\hat{y}) \right) \odot M_k(s_k) \right], \end{aligned} \tag{7.5}$$

and is normalized by the number of pixels contained in the mask $M_k(s_k)$. Note that when learning the generator G by maximizing $\mathcal{L}_a$ (Equation (7.1)), only the expectation over $\hat{y}$ is relevant, the other term being independent of G.

7.3.3 Semantic Matching Head

We argue that a perceptual loss commonly used to match the generated images with the target ones (*e.g.*, [238]) impose too strict requirements because an optimization in the pixel space would guide the model to yield these specific target images, while they should represent possible desired outputs only. Therefore, we introduce a semantic matching loss function to relax these constraints and instead match images in a semantic space which yields more diversity and less blurring in synthesized images.

For this, we add another head to the discriminator that predicts the semantic description s of the input image (y or $\hat{y}$). Specifically, as illustrated in Figure 7.2, for a structure s with K channels, the semantic matching head of the discriminator D_s outputs K maps each matching the correspondent constraint. The semantic loss is defined as a cross-entropy function between the upsampled outputs of this head (D_s) and the real semantic maps. This upsampling is necessary to match the size of the input semantic maps.

[2] extending the notation with $s_0 = s$.

7.3.4 Reconstruction Head

To regularize the training, some use a regression loss, *e.g.*, a L_1 loss [237]. However, it suffers from blurriness and lowering the diversity of generated images.

We introduce a novel reconstruction loss $\mathcal{L}_r$, as another head of the discriminator. This head acts as a regularizer: $\mathcal{L}_r = |f_{up}(D_r(y)) - y|$ where f_{up} stands for the upsampling function to match the image size. Note that this head is trained only on real images.

7.3.5 Stabilizing the Training

Employing the defined loss function and following the routine training process (training D with real and fake images and training G with fake images) leads to unstable training, which will be discussed in this section.

Take the joint distribution of training data as $p^*(x, s)$, the goal is to find an approximate joint distribution $p_\theta(x, s)$. The full objective function was defined in Equation (7.1) and Equation (7.2). For simplicity, we ignore the reconstruction loss here and therefore the objective function is as follows:

$$\underbrace{d(p^*(x), p_\theta(x))}_{ⓐ} - \underbrace{\mathbb{E}_{p^*(x,s)}[\log(q(s|x)]}_{ⓑ} - \underbrace{\mathbb{E}_{p_\theta(x,s)}[\log(q(s|x)]}_{ⓒ}, \tag{7.6}$$

where $d(p^*(x), p_\theta(x))$ is the Jensen-shanon divergence and $q(s|x)$ is the semantic matching head. The term ⓐ corresponds to common adversarial training loss that G tries to minimize it and D maximize. The terms ⓑ and ⓒ which correspond to real and fake images respectively, can be written as follows:

$$ⓑ = \mathbb{E}_{p^*(x)}[H_{p^*(x)}(s|x)] + \mathbb{E}_{p^*(x)}[KL(p^*(s|x)||q(s|x))], \tag{7.7}$$

$$ⓒ = \mathbb{E}_{p_\theta(x)}[H_{p_\theta(x)}(s|x)] + \mathbb{E}_{p_\theta(x)}[KL(p_\theta(s|x)||q(s|x))]. \tag{7.8}$$

To derive Equation (7.7), consider the following:

$$\begin{aligned}
&-\mathbb{E}_{p^*(x)}[H_{p^*(x)}(s|x)] = \mathbb{E}_{p^*(x)}[\log p^*(s|x)] \\
&= \mathbb{E}_{p^*(x,s)}[\log \frac{p^*(s|x)q(s|x)}{q(s|x)}] \\
&= \mathbb{E}_{p^*(x,s)}[\log q(s|x)] + \mathbb{E}_{p^*(x)}\mathbb{E}_{p^*(s|x)}[\log \frac{p^*(s|x)}{q(s|x)}] \\
&= \mathbb{E}_{p^*(x,s)}[\log q(s|x)] + \mathbb{E}_{p^*(x)}[KL(p^*(s|x)||q(s|x))].
\end{aligned} \tag{7.9}$$

If we replace p^* with p_θ, Equation (7.8) is derived.

For real images, Equation (7.7) is optimized. Its first term is zero and the second term makes $q(s|x)$ a good approximation of the real distribution $p^*(s|x)$.

For fake images, Equation (7.8) is optimized by two steps: (i) training G while D_s is frozen which pushes $p_\theta(s|x)$ towards $q(s|x)$, (ii) training D_s while keeping G frozen which pushes $q(s|x)$ towards $p_\theta(s|x)$. The effect of the second step is counter-intuitive. $q(s|x)$ is used as an intermediary distribution to help $p_\theta(s|x)$ be a variational approximation of $p^*(s|x)$ while this pushes $q(s|x)$ towards $p_\theta(s|x)$. In order to avoid that issue, the semantic matching head (D_s) is not trained on fake images leading to more stable training.

Another point in the training is the initialization of the semantic matching head in order to avoid the misguidance of the generator. In other words, in the beginning, G is not receiving any gradients from the semantic matching head since it is not trained well.

7.4 Experiments

We evaluate our model on two sets of experiments in different domains, showing the benefits of our proposed method on image synthesis from semantics: scene synthesis from segmentation maps (both car-view scenes and city buildings) and human synthesis from keypoints.

7.4.1 Scene Synthesis from Segmentation Maps

Datasets For the task of generating scene images from segmentation maps, we use two different datasets. The CMP Facades dataset [270] has 606 images of different resolutions of buildings with their $c = 12$ class semantic masks. We use the same split as [238], composed of 400 training, 100 validation and 106 test examples. Cityscapes [271] is a dataset of road scenes, with $2,975$ images in training and 500 in validation. Semantic annotations are segmentation maps in $c = 19$ classes. All images of those datasets are resized to a resolution of 256×256.

Implementation details We use the same generator as [238] and modify their two discriminators (in two different resolutions), *i.e.*, the last convolution layer (contains 512 kernels of size $4 \times 4 \times 1$) is replaced by three convolution layers, and each outputs multiple feature maps instead of a single feature map (contains 512 kernels of size $4 \times 4 \times ((c+1)+c'+3)$). The $c+1$ feature maps are related to the coarse-to-fine adversarial head, and c' and 3 are for semantic matching and the reconstruction heads, respectively. Similar to them, we replaced the LS-GAN loss in Equation (7.3) with the hinge loss. λ_s and λ_r are assigned to 1.0. Learning is conducted with Adam optimizer with a learning rate of 0.0002, momentum parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$ for 200 epochs. In the first 100 epochs, the generator is not trained with semantic matching and reconstruction heads, but for the next 100 epochs, it is trained by all heads. Moreover, we linearly decay the learning rate to 0 from epoch 100 to 200. Finally, the batch size is 32, and the hardware that we used contains 4 32GB V100 NVIDIA GPUs.

Semantic masks $M_k(s_k)$ are simply the downsampled segmentation masks defined by s_k. We keep the perceptual loss and feature matching loss, which were used in the baseline. Results of SPADE baseline [238] are obtained by re-training their model with their hyper-parameters publicly available.

Qualitative results Qualitative results are shown in Figure 7.3 for Cistyscapes and in Figure 7.4 for CMP Facades. Having the semantic matching head and different feature maps, each focusing on a specific object, could generate more semantically consistent details, *e.g.*, the windows and balconies are less blurry and with more details for the facades. By giving equal weight to all classes, our generator is able to better synthesize small objects that have few pixels or that are less frequent, such as doors in Facades, and buses, bikes, trains, and baby strollers in Cityscapes. The buildings are cleaner, and humans are more visible.

Quantitative results We report Fréchet Inception Distance (*FID*) [272], which compares the statistics of generated and real images. The results are presented in the second column of Table 7.1 and Table 7.2.

Similar to previous works [238], [256], a pre-trained segmentation network is used to see how well the predicted semantic masks match the ground truth input. In this study, Dilated Residual Network (DRN) [273] is used. The pre-trained models are obtained from their publicly available code. Note that we resize its input images to the resolution of 512×256. The quantitative evaluations and the comparison with previous works are presented in Table 7.1. The results show an improvement in all per-pixel accuracy, per-class accuracy, and mean-IOU.

In order to have a visual fidelity comparison against previous works, the Amazon Mechanical Turk (AMT) was used. The workers are given a semantic input mask and the outputs of the methods and are asked to choose the image which is more matched to the mask and is more realistic. The experiment consists of 500 questions, and each question is carried out by five

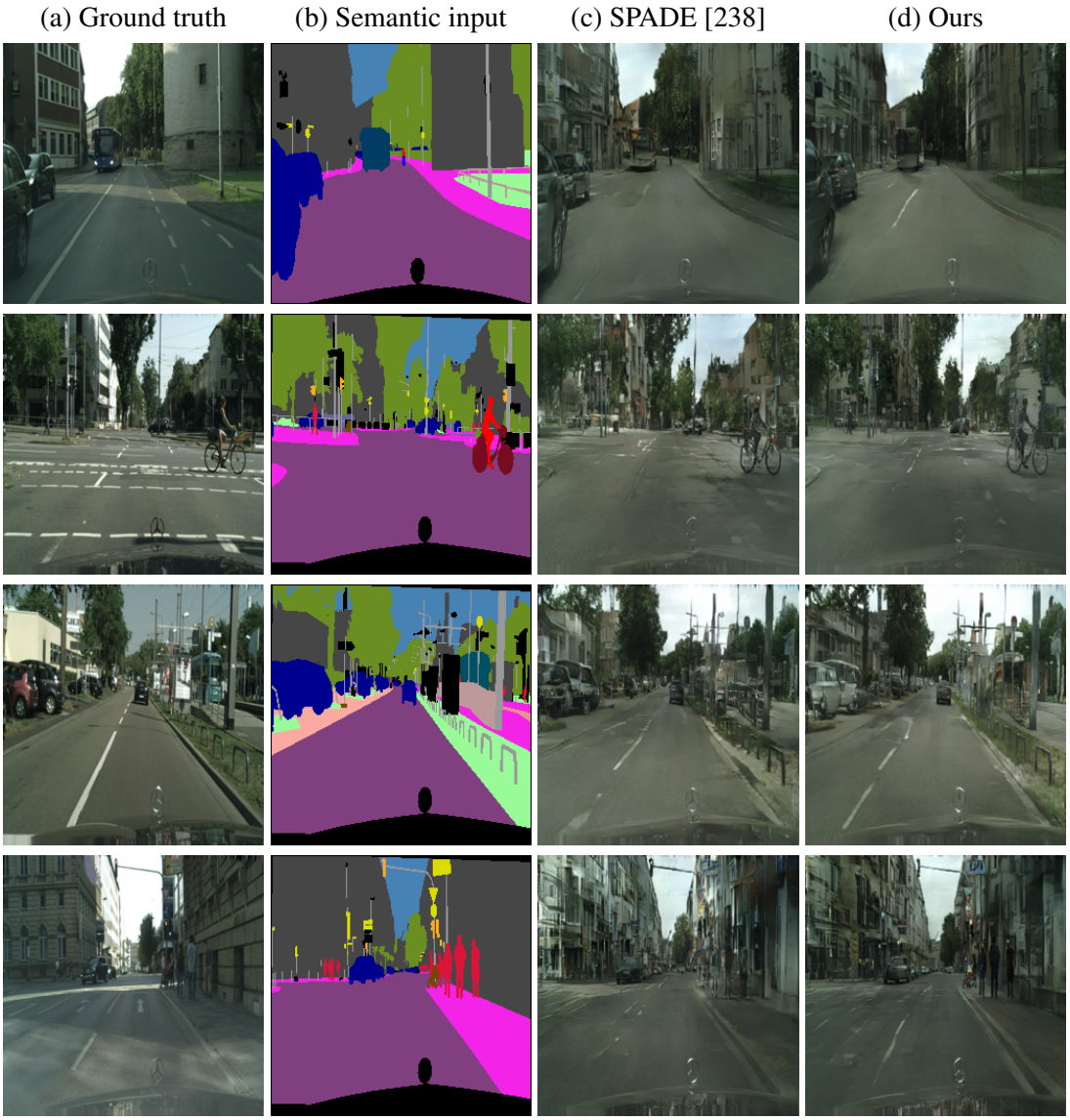


Figure 7.3: Qualitative results of image synthesis on Cityscapes dataset. Column (a) represents the ground truth. Its semantic map is shown in column (b). The results of the SPADE baseline using their pre-trained models are shown in column (c) followed by our proposed method in column (d).

different workers without any time limitation. The results can be found in the last column of Table 7.1. It shows that users preferred our results more than the baseline.

Evaluating on other baselines To show the generalization of our approach, we also applied the same procedure on pix2pixHD [234] and recently presented ASAPNet [274]. The results are in Table 7.1. To do that, we use the same generators as theirs and only modify their discriminators as

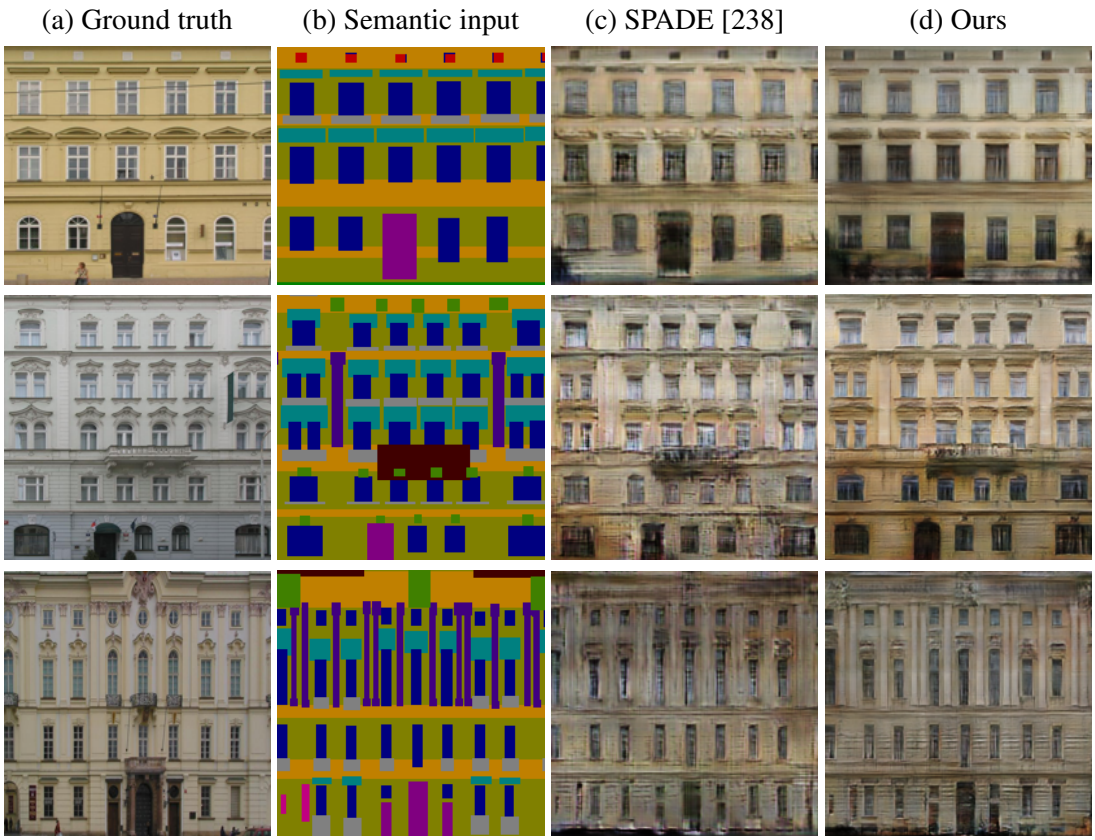


Figure 7.4: Qualitative results of image synthesis on CMP Facades dataset. Column (a) represents the ground truth. Its semantic map is shown in column (b). The results of SPADE baseline using their pre-trained models are shown in column (c) followed by our proposed method in column (d).

Models	Cityscapes [271]				
	FID ↓	mIOU ↑	pixel ↑	class ↑	user pref.
Pix2pix [237]	79.1	28.5	67.2	29.0	-
Pix2pixHD [234]	67.8	35.8	83.9	43.5	-
Pix2pixHD + Ours	55.8	44.4	89.2	52.7	-
ASAPNet [250]	69.2	29.6	77.2	35.1	-
ASAPNet + Ours	57.3	42.1	88.6	50.1	-
SPADE [238]	56.8	47.0	90.1	54.7	40%
SPADE + Ours	**50.8**	**55.9**	**92.3**	**64.2**	**60%**

Table 7.1: Quantitative evaluation of scene synthesis on Cityscapes dataset. Image quality: FID (the lower the better). Segmentation performance: mIOU, pixel and class accuracies (the higher the better). User preference study: the numbers show how much people preferred that method.

Models	CMP Facades [270]	
	FID ↓	user preference
SPADE [238]	121.4	29%
SPADE + Ours	**107.3**	71%

Table 7.2: Quantitative evaluation of scene synthesis on CMP Facades dataset. Image quality: FID (the lower the better). User preference study: the numbers show how much people preferred that method.

described. We again observe how impactful our method is. This modification of the discriminator of ASAPNet can improve their synthesized images without penalizing their speedup in inference since the discriminator is not used in inference time.

7.4.2 Human Synthesis from Keypoints

Datasets For the task of generating human images in a given pose (described as keypoints) with the same appearance as a source image, we validate our approach on the DeepFashion dataset [275]. This dataset includes $52,712$ clothing images with diverse person poses at the resolution of 256×256. Similar to [259], $200,000$ pairs of the same person-clothes with two different poses are used. We followed the same train/test split. This dataset does not have pose information labels. To obtain semantic annotations, a pre-trained pose detector [276] with $K = 18$ keypoints is used. Results of Deformable GAN baseline [239] are obtained from their publicly available code.

Implementation details The structure of G is identical to Deformable GAN [239]. We took both G and D from the baseline and modified its D. Learning uses the same hyper-parameters as in Section 7.4.1, but for 100 epochs. Semantic masks $M_k(s_k)$ are obtained as gaussians centered on the keypoints with a variance of $\sigma = 6$. Note that there is an extra encoder before the generator in this setting, which encodes the image appearance. This embedding vector is concatenated with the pose embedding and is fed to the decoder of the generator.

Qualitative results The results on the DeepFashion dataset are available in Figure 7.5. We observe that our discriminator adds more details, especially on faces and hands, without penalizing other parts. These details are even more visible in high-resolution images.

Quantitative results Quantitative results are presented in Table 7.3, where our proposed model achieves more realistic results in terms of FID than Deformable GAN. This shows that overall, our generated images are closer to the real ones. We have also performed a user preference study. We followed the same settings as the previous experiment.

Figure 7.5: Qualitative results of human image synthesis on DeepFashion dataset. The source and target images are in columns (a) and (b) respectively. The baseline, Deformable GAN, is shown in column (c) followed by our proposed method in column (d).

Models	DeepFashion [275]	
	FID ↓	user preference
Deformable GAN [239]	125.25	23%
Deformable GAN + Ours	**106.27**	**77%**

Table 7.3: Quantitative evaluation of human image synthesis on DeepFashion dataset.

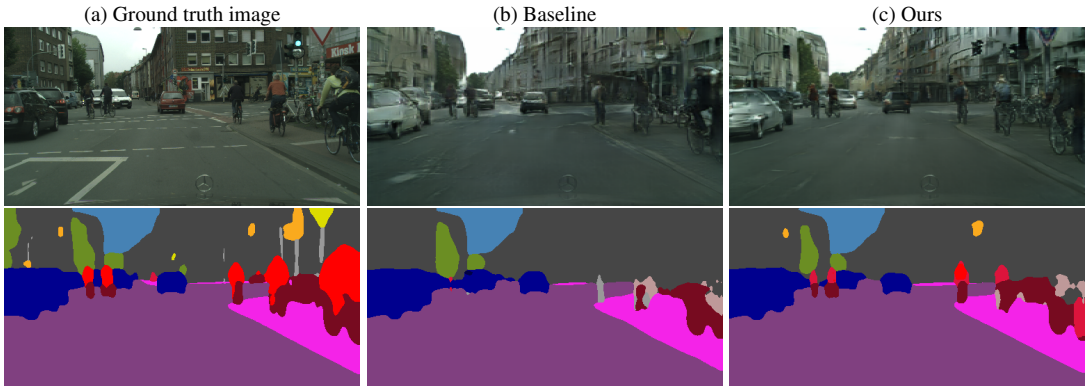


Figure 7.6: Comparison of our model vs the baseline in terms of matching the condition semantic map. The inputs of the semantic segmentation model are shown in the first row, and its outputs are in the second row. We show the ground truth image (a), the synthesized image of the baseline (b) and ours (c). Ours respect the condition semantic map more.

7.4.3 Ablation Study

In this section, an ablation study is provided to analyze the behavior of our model. For the ablation study, we consider the scene synthesis task on Cityscapes and compare it to the above-defined

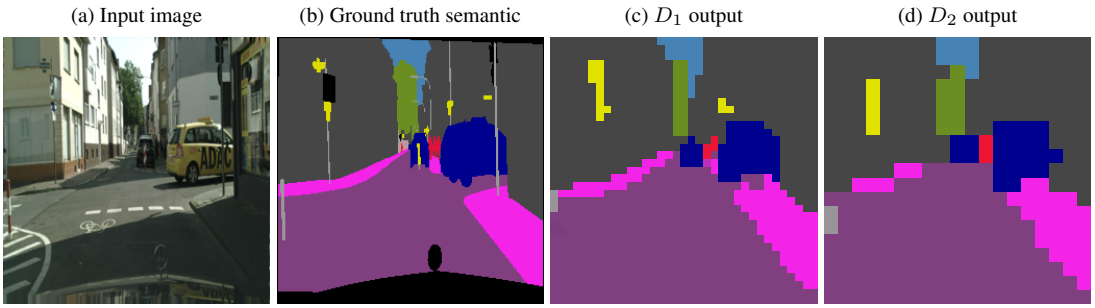


Figure 7.7: Visualization of the semantic matching head outputs. The input of the discriminator is shown in (a) followed by the ground truth image (b). In the baseline, there are two discriminators for different scales and we output both of them (c, d) which (d) is in lower resolution.

baseline [238].

The effect of the perceptual loss In this experiment, we want to observe the impact of the perceptual loss (extracted by a pretrained VGG network). As Table 7.4 shows, the performance of the baseline highly depends on the perceptual loss. However, the proposed method outperforms the baseline even without that loss. This proves that the proposed approach of using a multi-task semantic matching head could achieve better performance. Still, the perceptual loss is useful in improving the fidelity of images by adding some textures.

The effect of each head The analysis of each head is demonstrated in Table 7.5. It shows that all three heads are effective. The semantic matching head, with the help of the reconstruction task, has the best performance in segmentation metrics. Adding the coarse-to-fine adversarial head can effectively improve the quality of the images leading to a better FID.

Investigating the performance of the semantic matching head In Figure 7.6, the performance of the segmentation model on the synthesized images of our model is compared with the baseline. Humans, bikes, and traffic lights are clearly more detectable in ours. We also visualize the semantic matching head output to see whether it correctly does the semantic extraction. To do that, the output of the discriminator for an image at evaluation time is depicted in Figure 7.7. The baseline has two discriminators in different resolutions, and we show the outputs of both.

The effect of increasing the capacity of the network As previously mentioned, the added capacity to the architecture of D is minimum. Only the last convolution layer of D is modified, and instead of outputting a single feature map, it outputs multiple feature maps. The number of parameters for D has increased by approximately 10%, and is fixed for G. In one experiment, we increase the number of channels in the last layer of D without inducing our approach. Thus, the

Models	Cityscapes [271]			
	FID ↓	mIOU ↑	pixel ↑	class ↑
SPADE [238] w/o VGG	63.0	42.1	88.6	49.7
SPADE [238] w/ VGG	56.8	47.0	90.1	54.7
Ours* w/o VGG	56.2	54.8	92.2	62.4
Ours* w/ VGG	**52.6**	**56.3**	**92.3**	**63.9**

Table 7.4: The comparison of the effect of perceptual loss in the baseline and ours. Here, our model has the semantic matching head, the reconstruction heads and a simple (not coarse-to-fine) adversarial head.

Models	Cityscapes [271]			
	FID ↓	mIOU ↑	pixel ↑	class ↑
SPADE [238]	56.8	47.0	90.1	54.7
Ours (sem)	53.4	55.9	92.2	63.7
Ours (sem + rec)	52.6	**56.3**	**92.3**	63.9
Ours (c2f)	52.7	45.2	89.7	52.5
Ours (c2f + sem)	51.9	55.2	92.2	62.8
Ours (c2f + sem + rec)	**50.8**	55.9	**92.3**	**64.2**

Table 7.5: Ablation study on the discriminator heads. c2f: coarse-to-fine adversarial head, rec: reconstruction head and sem: semantic matching head

Models	Cityscapes [271]			
	FID ↓	mIOU ↑	pixel ↑	class ↑
SPADE [238]	56.8	47.0	90.1	54.7
SPADE + 10% capacity	57.9	44.5	89.8	52.1
Ours	**50.8**	**55.9**	**92.3**	**64.2**

Table 7.6: The effect of adding 10% more capacity to the baseline vs ours.

number of parameters of D is increased by 10%. As Table 7.6 shows, simply adding capacity without our shared representation could not improve the performance. Moreover, note that the added capacity only affects the training time, and there is no overhead at inference time.

7.5 Conclusions

Image synthesis has a significant impact on modern transportation. However, current models lack sufficient photorealism. In this thesis, we have presented a new semantically-aware discriminator to better guide the training of conditional Generative Adversarial Networks (GANs). We have shown that augmenting the discriminator's task with pixel-level content understanding improves the classification of real and fake images. Our contributions are generic and applicable to any

generator network for image synthesis. Future work can focus on extending our approach to enforce temporal consistency in video synthesis.

8 Conclusions

This thesis has provided a broad exploration into the pressing challenges that revolve around the deployment of autonomous systems. With a focus on deep generative models and their applications in autonomous driving, we have ventured into the fields of trajectory and pose forecasting, as well as image synthesis for simulation applications. Here, we summarize our findings, and propose possible future directions of research.

8.1 Findings

Chapter 2 evaluated trajectory forecasting models with the new perspective of adversarial attack and revealed that state-of-the-art prediction models forecast socially unacceptable trajectories, *i.e.*, with collisions. By employing adversarial training with our approach, we were able to improve the social understanding of these models and enhance their robustness against adversarial attacks.

Chapter 3 proposed an innovative approach, *i.e.*, adversarial scene generation to probe the robustness of vehicle trajectory forecasting. Through the generation of physically plausible scenes that lead to off-road predictions, we were able to expose the vulnerabilities of forecasting models and develop strategies to bolster their robustness.

Chapter 4 proposed the use of visual cues that humans naturally exhibit to communicate their intention in trajectory forecasting, integrating these diverse signals into a universal model by applying the notion of prompts. This led to an enhancement in the accuracy of human trajectory predictions, indicating the potential of augmenting input trajectory data.

Chapter 5 targeted human pose forecasting by first introducing a comprehensive open-source library for a unified and consistent evaluation in the field. Then, by modeling uncertainty and quantifying it effectively, we were able to boost performance and engender greater trust in human pose forecasts.

Chapter 6 proposed a generic diffusion-based approach that can handle noisy observations in

pose forecasting, treating missing elements as noise to be denoised. Our model showed superior performance in both noiseless and noisy environments and could enhance any state-of-the-art forecasting model by repairing their inputs and refining their outputs.

Finally, in Chapter 7, we ventured into image synthesis for simulation applications. In this context, a novel semantically-aware discriminator was introduced, significantly enhancing the training of conditional Generative Adversarial Networks. This novel approach led to the production of high-fidelity images that can accurately simulate real-world scenarios, providing a valuable tool for the training and testing of autonomous driving systems.

8.2 Future Research Directions

Within the vast domain of deep generative models applied to autonomous driving, numerous territories remain uncharted. In this section, we delineate several prospective avenues warranting future exploration.

A Unified Diffusion Model for In-distribution Trajectory Forecasting: One of the issues with trajectory forecasting models is the lack of generalizability, *i.e.*, their inability to operate across new datasets due to distribution shifts. For instance, a model trained on a synthetic dataset may not perform optimally on a real-world dataset without necessary adaptations. Moreover, there can be rare scenarios where it is essential to ensure that trajectory forecasts align within an expected distribution, a crucial factor for safety. Could diffusion models serve as tools to align various distributions to a standard distribution?

Recent advancements in adversarial purification signal the potential of diffusion models in mitigating some variations [277]. A potential avenue for future research could involve the use of diffusion models to process input trajectories, irrespective of their source – simulated or real-world datasets with varying distributions. This approach would align all distributions, ensuring that prediction models consistently receive trajectories conforming to the desired distribution, thereby enhancing their overall performance. Another research direction could be to shift the outputs of trajectory forecasting models using diffusion models toward the desired distribution. As an extension of what we discussed in Chapter 6, the learned diffusion model could serve as a tool to promote safer outputs in line with the desired observed distribution.

Augmenting Inputs for Trajectory Forecasting: Currently, the effectiveness of trajectory forecasting models is partly hindered due to their dependence on upstream detection models. Indeed, the adoption of end-to-end methods has increased recently [278]. By leveraging our Social-Transmotion framework, there is potential to extend our universal model to forecast trajectories directly from raw sensor data such as images or LIDAR scans. Furthermore, this approach presents the opportunity to evaluate and amplify the performance of trajectory forecasting models that consider scenes as pivotal inputs, by enriching input data.

Rethinking Outputs in Motion Forecasting: So far, we have treated trajectory and pose

forecasting as separate problems. Yet, their intertwined nature implies that errors in one could substantially impact the other. Our preliminary experiments showed promising outcomes when both aspects are considered jointly. In light of this, it is worthwhile for future research to pursue the development of a comprehensive motion forecasting model. Such a model, an extension of our Social-Transmotion framework, could simultaneously and realistically forecast trajectory, pose, and several other attributes. Specifically, one can adopt a multi-task learning strategy for Social-Transmotion and incorporate our proposed diffusion-based approach.

Study of Real-World Noise Distribution: In this thesis, we explored the effect of various noises–random noise, adversarial perturbations, and incomplete observations–on the reliability and performance of autonomous driving systems. A key factor affecting the reliability and performance of autonomous driving systems in real-world deployment is the prevalence of various noise sources, including sensor disturbances and interruptions. A comprehensive understanding of real-world noise distributions could provide invaluable insights into strengthening the robustness of our motion forecasting models in real-world scenarios.

Extrapolative Generative Models: The capability of current generative models is often restricted to generating data closely mirroring the training set, thereby limiting their extrapolation ability. It is worthwhile to investigate models capable of extrapolating beyond observed training data and generating out-of-distribution samples, especially in high-dimensional spaces. Such an approach could broaden the application spectrum for synthesis models, for instance, in generating images missing from the training data.

A Supplementary Materials of Socially-aware Trajectory Forecasting

In this chapter, we begin by presenting additional qualitative results of the attacks. We then provide a visualization of the training process and describe the detailed hyperparameters used. Lastly, we offer a justification for the attention weights employed in the model.

A.1 Additional Qualitative Results

The guidelines for the visualizations in this appendix are mentioned in Figure A.1.

In Chapter 5, we have shown some results of the proposed attack. Here, we add some more successful results of the proposed attack on the main model (D-Pool) in Figure A.2. Moreover, Figure A.3 shows some failure cases of the proposed attack. Although the attack tried to make a collision, this amount of perturbation limit was not sufficient. By increasing the allowed perturbation limit, possibly a collision occurs even in these scenarios.

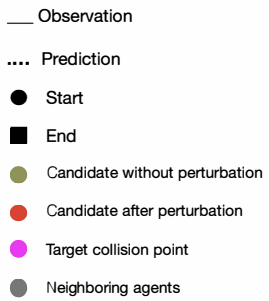


Figure A.1: The visualization guide of the supplementary.

A.2 Visualization of the Training Process

Here, we aim to depict the evolution of the training process.

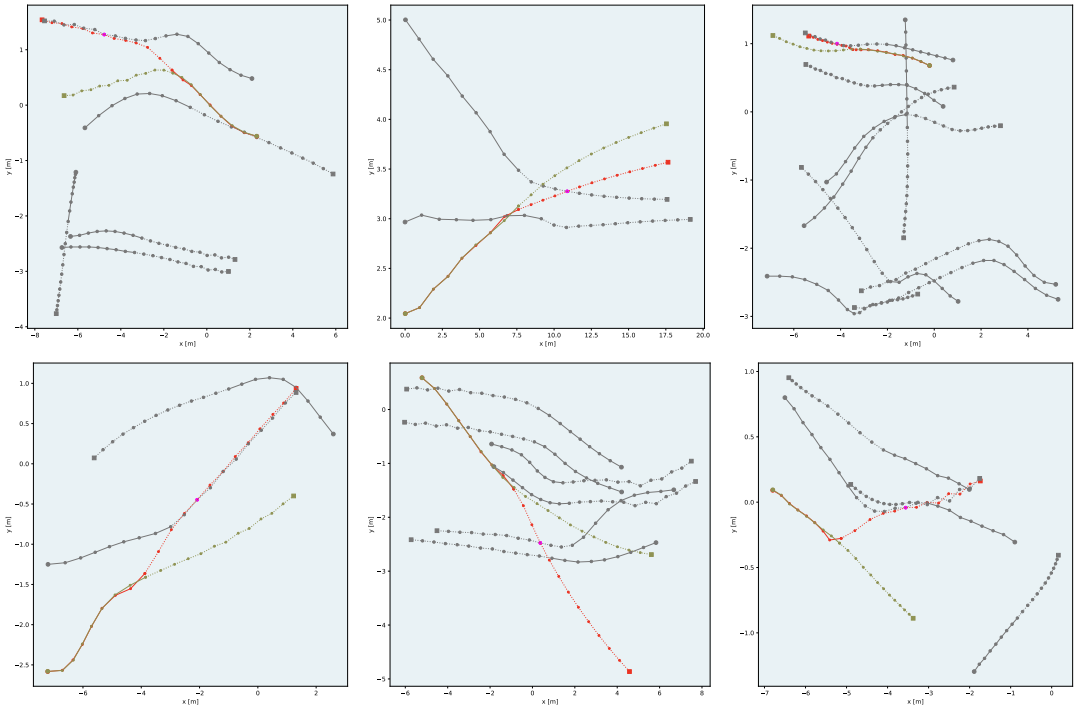


Figure A.2: More qualitative results of our attack on D-Pool. The results show the qualitative outcomes of proposed attack. The P-avg values are 0.024m, 0.010m, 0.013m, 0.017m, 0.012m and 0.016m, starting from top-left figure to bottom-right.

In the first figure of Figure A.4, the relative attention weights W are displayed. For ease of interpretation, we represent the maximum element of W in each iteration by black, with other elements shaded accordingly between black and white. A darker shade denotes a higher attention from the network on that specific agent-timestep. As training progresses, W undergoes updates, and eventually, the attack settles on a specific agent-timestep to cause a collision.

In the right figure, the predictions of the trajectory prediction model in different iterations are observed. The observed changes are a result of the alteration in attention weights as the observation sequence is subtly perturbed by the attack. The model tries to make a collision with a small perturbation size (P-avg) which is achieved in the 100-th iteration.

A.3 Hyper-parameters

The training details are as follows. We perform the common steps for PGD [60] with the maximum size of 0.2m for the perturbation of each observation point. The complete list of hyperparameters and the learning rates are in Table A.1. Moreover, we linearly decay the learning rate to 0 when the loss is not improving. We use the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$.

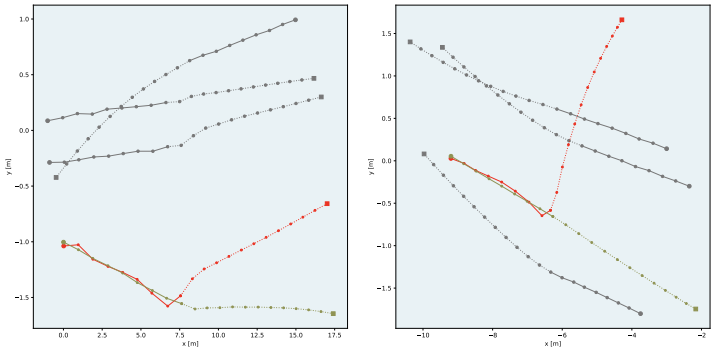


Figure A.3: Some failure cases of our attack on D-Pool. Despite the extensive variation in predicted trajectories for perturbed scenarios (compared to the original predictions), yet the attack fails to make collisions. The P-avg values are 0.042m and 0.049m for the left and right figures.

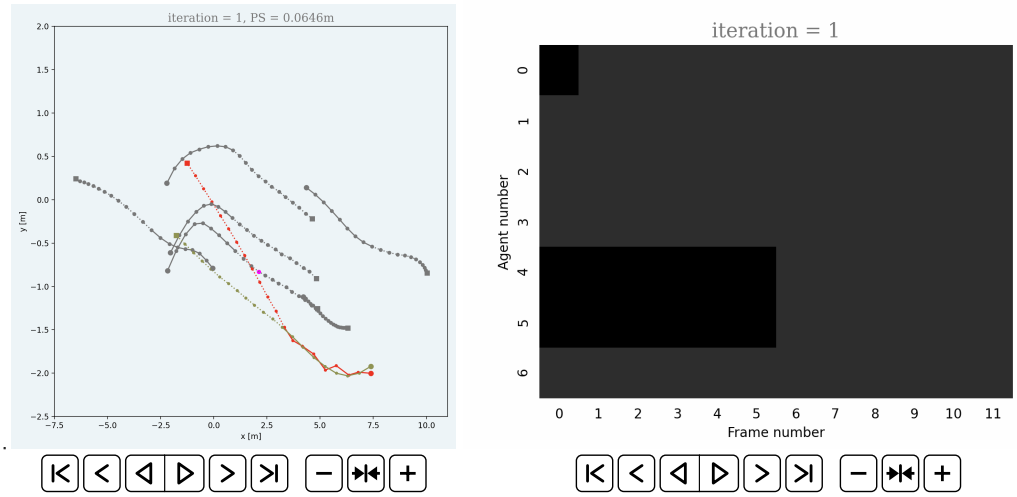


Figure A.4: Two animations showing the model's changes across iterations. It is best viewed using Adobe Acrobat Reader.

Method	Learning rate α	λ_r	λ_w
hard attention	$1e^{-1}$	0.1	-
soft attention	$1e^{-2}$	0.1	0.5

Table A.1: List of hyper-parameters in S-ATTack.

A.4 The Proof of Regularization of Attention Weights

In Chapter 2, we introduced a new term $\|W\|_F$ which is maximized in the optimization problem. Here, we fully describe why this term is useful in attaining the convergence point of W faster. Let us start with the constraints on W:

$$\sum_{j,t} w_{j,t} = 1, \quad w_{j,t} \geq 0 \quad \rightarrow \quad 0 \leq w_{j,t} \leq 1, \quad [w_{j,t}]^2 \leq w_{j,t} \tag{A.1}$$

Thus, the regularization of attention weights will be as follows:

$$\|W\|_F = \sqrt{\sum_{j,t} [w_{j,t}]^2} \leq \sqrt{\sum_{j,t} w_{j,t}} = 1 \tag{A.2}$$

The maximum of the above formula happens when the equality holds and that needs all $[w_{j,t}]^2 = w_{j,t}$. Therefore, $w_{j,t} \in \{0, 1\}$. Noticing the sum constraint $\sum_{j,t} w_{j,t} = 1$, our algorithm makes one element of W matrix as 1 and the rest as 0 in the final convergence point. This means that this term guides W in the desired direction.

B Supplementary Materials of Scene-aware Trajectory Forecasting

B.1 Additional Qualitative Results

1. **Real-world retrieval images.** We show more real-world examples for both cases where the trajectory prediction model fails and succeeds in Figure B.1.

2. **More generated scenes.** Figure B.2 provides more visualizations for the performance of the baselines in our generated scenes.

3. **Noise in the drivable area map.** The models predict near perfect in the original dataset with HOR of less than 1%. Our exploration shows that most of the 1% failed cases are due to the annotation noise in the drivable area maps of the dataset and the models are almost error-free with respect to the scene. Some figures are provided in Figure B.3.

4. **Animated Illustrations.** Refer to Figure B.4 for animated demonstrations showcasing the model's performance as the scene undergoes a smooth transformation. We observe that in some cases the model fails and in some succeeds.

B.2 Additional Quantitative Results

Excluding trivial scenes: In this part, we remove some trivial scenes, i.e., the scenes that fooling is near impossible, e.g., the scenes with zero velocity. Excluding them, we report in Table B.1 and compared to Table 3.1, the off-road numbers substantially increase.

Exploring black box algorithms: In Chapter 3, we mentioned that we used a brute-force approach for finding the optimal values as the search space is not huge. Here, we investigate different block box algorithms for the search. The results of applying different search algorithms are provided in Table B.2. They cannot overcome the brute-force approach because of their bigger search spaces (the continuous space instead of the discrete space) and the large required computation time.

(a) Paris location (b) New York location (c) Hong Kong location

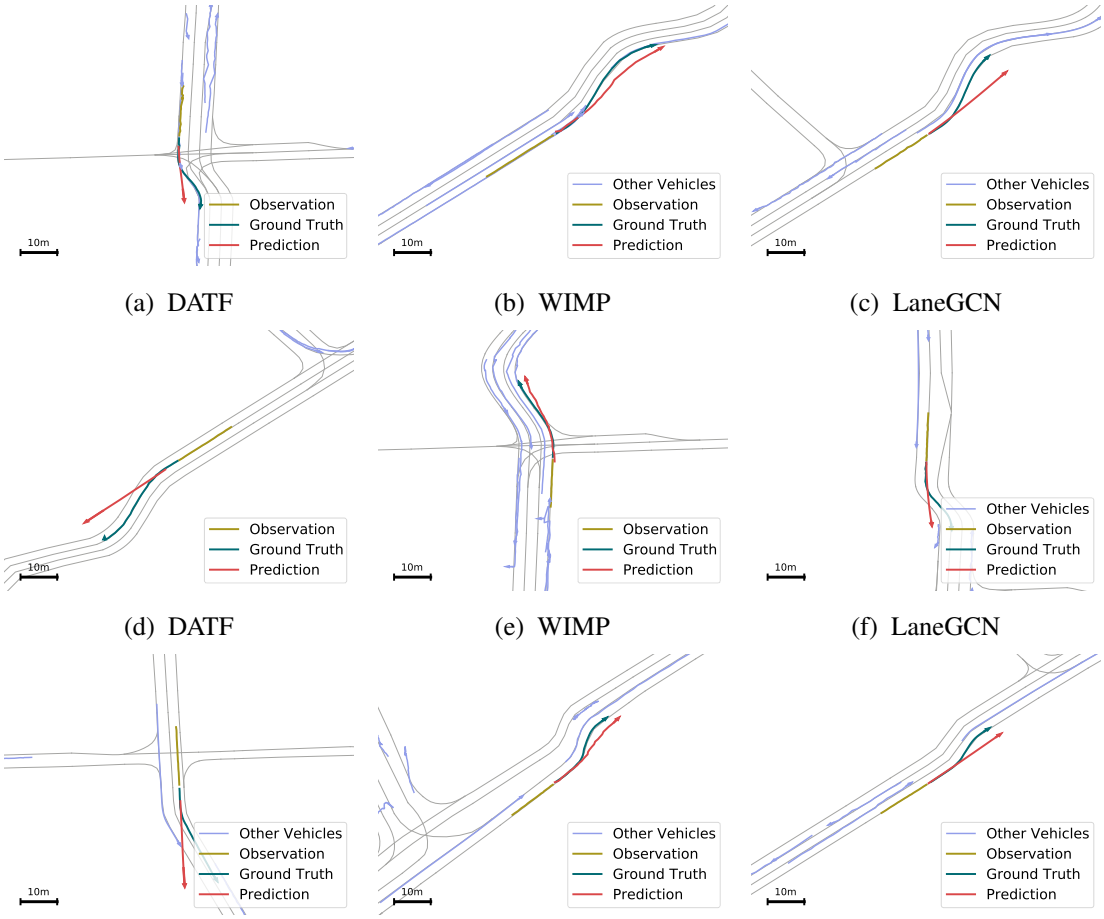
(d) New Mexico location (e) Hong Kong location (f) Paris location

(g) New York location (h) New Mexico location (i) New York location

Figure B.1: Retrieving real-world places using our real-world retrieval algorithm. We observe that the model fails in Paris (a), New York (b), Hong Kong (c) and New Mexico (d). The model also successfully predicts in the drivable area in the remaining figures.

B.3 Overall Algorithm

In this section, we demonstrate the overall algorithm for the chosen search method. The pseudo-code of the algorithm for generating a scene is shown in Algorithm 2. The goal is to generate the scene S^* for a given scenario x, a, S and predictor g. The process is called for k_{max} iterations. In each iteration, we start with selecting a transformation function (L. 3). Then, the transformation

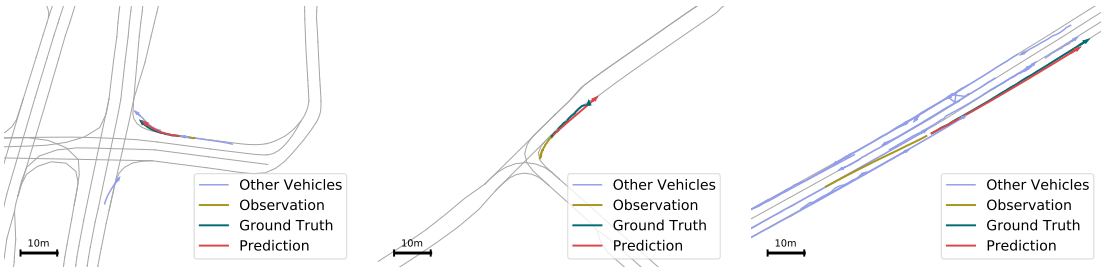


(a) DATF (b) WIMP (c) LaneGCN

(d) DATF (e) WIMP (f) LaneGCN

(g) DATF (h) WIMP (i) LaneGCN

Figure B.2: The qualitive predictions of different models in some generated scenes. All models are challenged by the generated scenes and failed in predicting in the drivable area.



(a) (b) (c)

Figure B.3: Some examples showing the noise in the drivable area map. All these predictions were considered as off-road because of an inaccurate drivable area map.

function generates the corresponding scene (L. 4). After that, the observation trajectory is scaled to ensure the feasibility of the scenario (L. 5). Next, the prediction of the model in the new

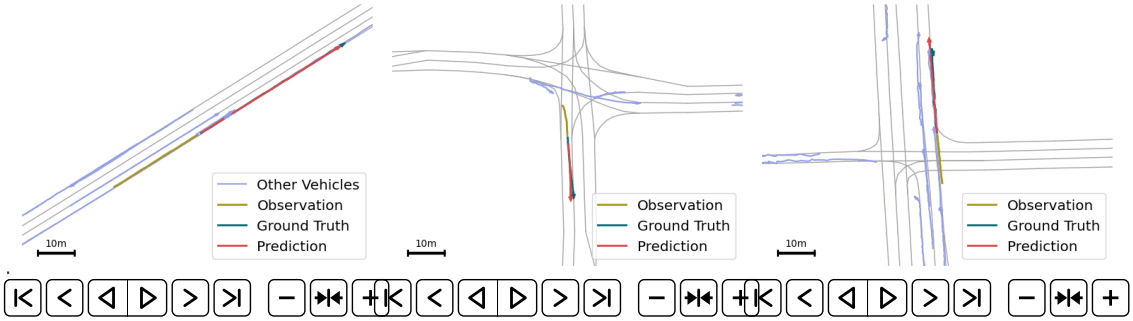


Figure B.4: The animations showing the changes of the model's predictions in different scenes. It is best viewed using Adobe Acrobat Reader.

Model	Original	Generated (**Ours**)			
		Smooth-turn	Double-turn	Ripple-road	All
	SOR / HOR	SOR / HOR	SOR / HOR	SOR / HOR	SOR / HOR
DATF [72]	1 / 2	44 / 92	43 / 91	50 / 95	51 / 99
WIMP [113]	0 / 1	30 / 80	23 / 71	29 / 77	31 / 82
LaneGCN [15]	0 / 1	23 / 65	32 / 75	34 / 77	37 / 81
MPC [115]	0 / 0	0 / 0	0 / 0	0 / 0	0 / 0

Table B.1: Comparing the performance of different baselines in the original dataset scenes and our generated scenes after removing trivial scenarios. SOR and HOR are reported in percent and the lower represent a better reasoning on the scenes by the model. Numbers are rounded to the nearest integer.

scenario is computed and used to calculate the loss (L. 6, L. 7). The best-achieved loss determines the final generated scene.

B.4 Generalization to Rasterized Scene

In Chapter 3, we assumed S is in the vector representation, *i.e.*, it includes x-y coordinates of road lanes points. In the case of a rasterized scene, an RGB value is provided for each pixel of the image. Therefore, it is the same as the vector representation unless here we have information (RGB value) about other parts of the scene in addition to the lanes. Hence, the transformation function can be applied directly on all pixels of the image. In other words, in image representation, s is the coordinate of each pixel which has an RGB value and $\hat{s}$ represents the new coordinate with the same RGB value as s.

Optimization method	on LaneGCN [15] SOR / HOR	GPU hours
Baysian [108], [109]	13 / 40	17.5
GA [111]	14 / 45	25.0
TPE [112]	14 / 45	12.1
Brute force	23 / 66	4.2

Table B.2: Comparing the performance and computation time of different optimization algorithms in the generated scenes.

Algorithm 2: Scene search method

Input: Sequence h, Scene S, Predictor g, Surrounding vehicles a, Transformation set f, Number of iterations k_{max}

Output: Generated scene S^*

```
Initialize l* ← 1
for k = 1 to k_max do
    Choose a transformation function
    S̃ = [s̃] where s̃ ← Equation (3.2)
    Obtain h̃, ã from phys constraints Section 3.3.3
    z̃ = g(h̃, S̃, ã)
    Calculate l using Equation (3.7)
    if l < l* then
        S* = S̃
    end
end
```

C Supplementary Materials of Human Pose Forecasting with Uncertainty

Here, we provide supplementary materials to Chapter 5. They comprise an experiment that explores the uncertainties of various joints and priors, qualitative results, and ablation studies related to aleatoric uncertainty. Additionally, we include an evaluation of EpU across a wider range of actions, as well as a motion clustering analysis for epistemic uncertainty.

C.1 Aleatoric Uncertainty in Pose Forecasting

C.1.1 Study of Joints' Aleatoric Uncertainties

In Chapter 5, we observe that the uncertainty of joints increases over time. Another observation in our experiments is that different joints have different behaviors. For instance, Figure C.1 shows that hand joints have lower uncertainties compared to leg joints in the beginning of the forecasting as hands move less and are more predictable. However, toward the end of the forecasting, hands are more unpredictable, therefore have higher uncertainties compared to legs.

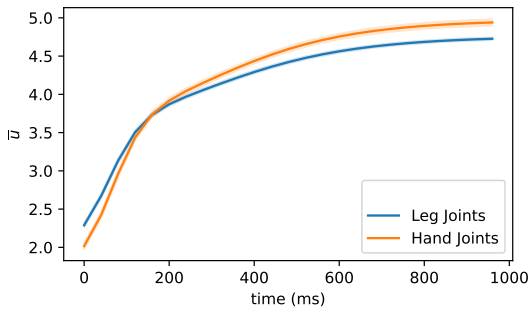


Figure C.1: Evolution of uncertainty of hands and legs over time. Hands' uncertainty is lower at short prediction horizon, but higher at longer prediction horizons.

C.1.2 More Comparison of Priors

In Chapter 5, we observed that using a prior can lead to similar aleatoric uncertainty than the unconstrained case, but with fewer learnable parameters and better stability. Here, we plot the learned aleatoric uncertainty of different prior functions in Figure C.2. We observe that all priors lead to the same general evolution over time which comes from the exponential behavior of error in time; however, Sig_5 matches the best.

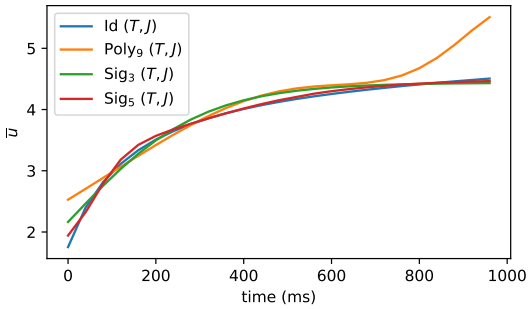


Figure C.2: The values of the learned aleatoric uncertainties for different priors trained on ST-Trans.

C.1.3 Additional Qualitative Results

Examples of forecast pose sequences are depicted in Figure C.3. We observe higher uncertainties for later time frames.

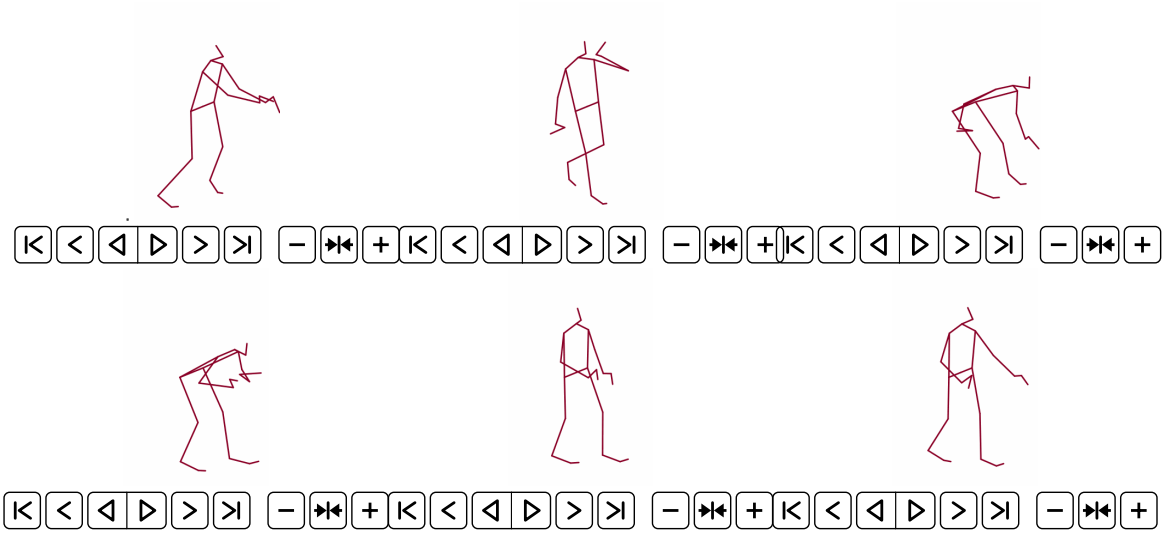


Figure C.3: Six animations showing different forecast pose sequences. Higher aleatoric uncertainty is shown with a lighter color. It is best viewed using Adobe Acrobat Reader.

#Layers	#Heads	#Channels	80 ms	160 ms	320 ms	400 ms	560 ms	720 ms	880 ms	1000 ms
6	8	64	10.4	**23.4**	**48.4**	**59.2**	**77.0**	**90.7**	**101.9**	**109.3**
6	4	64	10.6	23.8	49.3	60.3	78.3	91.9	103.0	110.2
6	10	60	**10.3**	23.5	48.8	59.7	77.5	91.0	102.2	109.4
6	8	32	10.5	23.8	49.4	60.1	78.1	91.8	102.9	109.9
6	8	80	12.0	25.9	51.7	62.2	79.3	92.4	103.4	110.2
4	8	64	10.4	23.5	48.9	59.8	77.7	91.3	102.3	**109.3**
8	8	64	10.5	23.7	48.8	59.6	77.5	91.1	102.4	109.7

Table C.1: Ablation studies of ST-Trans on Human3.6M [140] in MPJPE (mm) at different prediction horizons. The model shown in the first row is identified as the final one.

C.1.4 Ablation Studies

In Chapter 5, we introduced ST-Trans. Here, we ablate three main design choices (number of layers, number of attention heads, and number of channels) in Table C.1.

C.2 Epistemic Uncertainty in Pose Forecasting

C.2.1 Additional Evaluation Across Various Actions

Evaluating the quality of epistemic uncertainty is difficult due to the unavailability of ground truth annotations. In Chapter 5, we conducted an experiment on classifying walking-related and sitting-related actions. Further assessments on a wider array of actions are presented here, with results detailed in Table C.2 and ROC curves depicted in Figure C.4. The findings indicate that our method surpasses prior approaches in nearly all tested conditions. Specifically, as shown in Table C.2's final column, when our clustering and forecasting model was trained on all non sitting-related actions (13 actions) and evaluated for its ability to differentiate these from the remaining sitting-related actions (2 actions), our method achieved superior AUROC values and ROC curves compared to competing methods.

ID actions OOD actions	Walking Purchases	Walking TakingPhoto	Smoking Phoning	Smoking Sitting	Discussions Directions	w/o Sitting Sitting
Deep-Ensemble-3	0.83	0.80	0.51	0.69	0.54	0.68
Deep-Ensemble-5	0.86	0.80	0.54	**0.77**	0.58	0.68
MC-Dropout-5	0.80	0.80	0.48	0.65	0.51	0.54
MC-Dropout-10	0.83	0.82	0.49	0.67	0.52	0.55
Ours	**0.93**	**0.89**	**0.56**	0.71	**0.59**	**0.76**

Table C.2: AUROC for different sets of actions for different epistemic uncertainty methods. The actions on top indicate the training ID actions and the ones below them indicate the test OOD actions.

C.2.2 Motion Clustering

This section presents visualizations demonstrating the separability of our data points before and after the clustering process. In Figure C.5, we observe that raw data points are not well-separated. Training with only the reconstruction loss improves the separability, and finally, joint optimization of the reconstruction loss and deep clustering makes data clearly distinguishable in the learned feature space.

To derive EpU, we opted to use clustering in the representation space instead of alternative methods, such as action recognition models or the action labels of existing human pose datasets, e.g., Human3.6M. There are several differences between a motion and an action when dealing with human pose sequences: 1) the actions are limited, whereas motions can be much more varied; 2) multiple consecutive distinct motions usually constitute an action. Motion clustering is also generalizable to datasets without action labels and real-world settings. Here, we show three motion examples in Figure C.6 where the first two are from the *Purchases* action and the third one is from the *Walking* action. As can be seen, the last two motions are very similar, although they have different action labels.

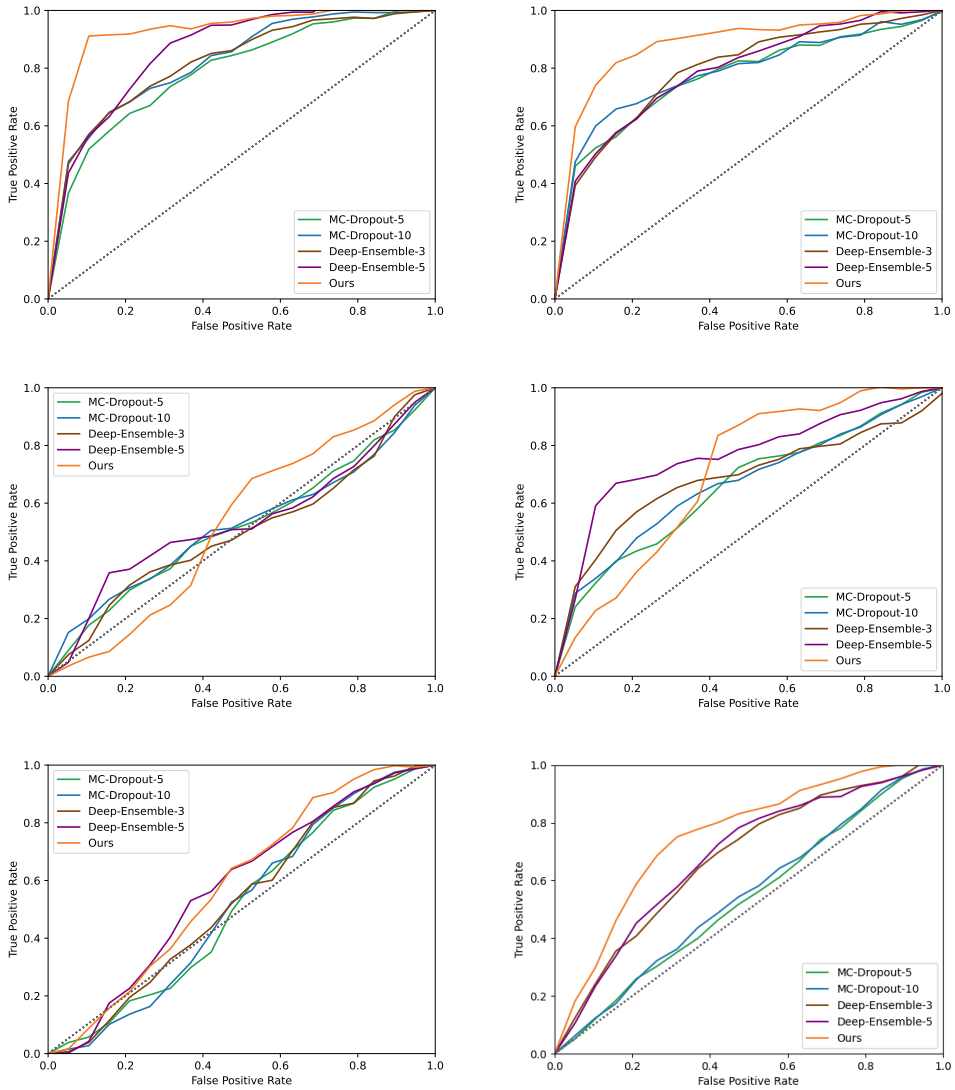


Figure C.4: ROC curve for a model trained on the first actions and tested on both first and second actions. The objective is to distinguish between these sets by utilizing uncertainty estimates. (a) Walking - Purchases (b) Walking - TakingPhoto (c) Smoking - Phoning (d) Smoking - Sitting (e) Discussion - Directions (f) w/o Sitting - Sitting

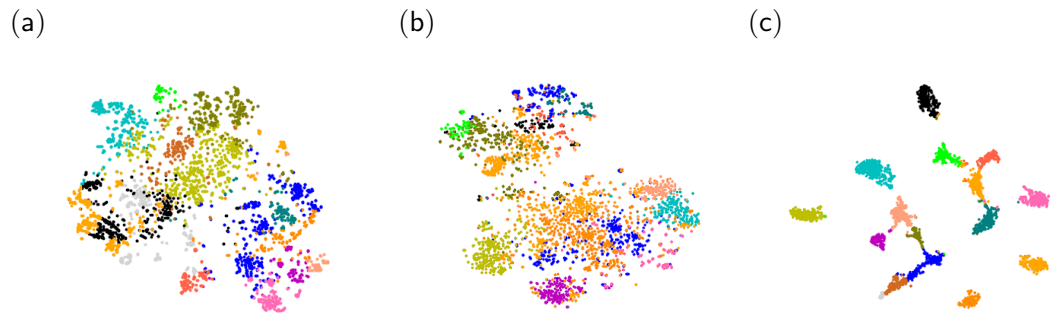


Figure C.5: t-SNE visualization of the test set of Human3.6M (a). The reconstruction loss helps in shaping the feature space (b). Joint optimization of the reconstruction loss and deep clustering leads to a clear distinction of data (c).

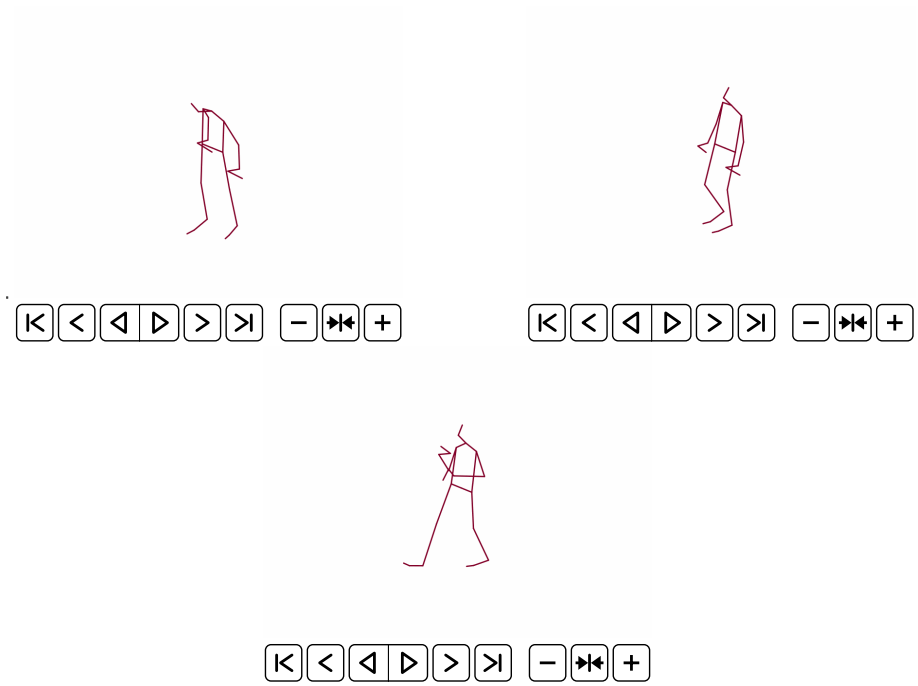


Figure C.6: Three motions corresponding to the action classes *Purchases* (at the top), and *Walking* (at the bottom). It is best viewed using Adobe Acrobat Reader.

D Supplementary Materials of Pose Forecasting in Noisy Observations

Here, we extend our comparisons in Chapter 6:

1. We compared our model's performance with the models that reported A-MPJPE upto different horizons in Table D.1. Our setting was changed to predict 25 frames given 10 observation frames on the Human3.6M dataset down-sampled to 25 fps with the subset of 22 joints (Setting-D), following the settings of [174]. The evaluation of the models was conducted on all actions except walking together. Our model outperforms those GCN-based models.

Model	80ms	160ms	320ms	400ms	560ms	1000ms
Zero-Vel	18.1	28.7	46.9	54.6	67.7	93.3
STSGCN [159]	10.2	17.3	33.5	38.9	51.7	77.3
GAGCN [174]	10.1	16.9	32.5	38.5	50.0	72.9
TCD (ours)	**7.4**	**14.0**	**27.7**	**33.9**	**44.7**	**66.5**

Table D.1: Comparison with deterministic models on Human3.6M [140] Setting-D in A-MPJPE (mm) at different prediction horizons.

2. We conducted another experiment to compare our model's performance with others that reported their results on Human3.6M Setting-E, as shown in Table D.2. In this setting, 25 frames are predicted given 10 observation frames down-sampled to 25 fps with the subset of 17 joints. Our model outperformed others, particularly in longer horizons.

3. In Table 6.2, we compared the performance of different models on Human3.6M [140] Setting-B. The detailed results on all categories are reported in Table D.3. We observe that in almost all categories, ours beats previous models.

Model	80ms	160ms	320ms	400ms	560ms	1000ms
Zero-Vel	17.1	31.9	54.8	63.8	78.3	100.0
LDRGCN [223]	10.7	22.5	45.1	55.8	–	97.8
MPT [279]	8.3	18.8	39.0	47.9	65.3	96.4
TCD (ours)	**8.3**	**18.8**	**37.8**	**44.9**	**55.9**	**76.9**

Table D.2: Comparison with deterministic models on Human3.6M [140] Setting-E in MPJPE (mm) at different prediction horizons.

Scenarios	Walking						Eating						Smoking						Discussion					
Model	80ms	320ms	560ms	720ms	880ms	1000ms	80ms	320ms	560ms	720ms	880ms	1000ms	80ms	320ms	560ms	720ms	880ms	1000ms	80ms	320ms	560ms	720ms	880ms	1000ms
Zero-Vel	33.9	109.8	145.9	154.4	150.7	140.2	16.5	55.3	81.3	94.4	100.7	102.1	17.3	57.1	80.3	91.4	98.1	101.1	24.5	76.8	108.7	123.5	131.5	135.3
Res. Sup.	23.2	61.0	71.6	72.5	76.0	79.1	16.8	53.5	74.9	85.9	93.8	98.0	18.9	57.5	78.1	88.6	96.6	102.1	25.7	80.0	109.5	122.0	128.6	131.8
convSeq2Seq	17.7	56.3	72.2	77.2	80.9	82.3	11.0	40.7	61.3	72.8	81.8	87.1	11.6	41.3	60.0	69.4	77.2	81.7	17.1	64.8	98.1	112.9	123.0	129.3
LTD	12.3	39.4	50.7	54.4	57.4	60.3	7.8	31.3	51.5	62.6	71.3	75.8	8.2	32.8	50.5	59.3	67.1	72.1	11.9	55.1	88.9	103.9	113.6	118.5
HRI	10.0	**34.2**	47.4	52.1	55.5	58.1	6.4	**28.7**	50.0	61.4	70.6	75.7	7.0	29.9	**47.6**	56.6	64.4	69.5	10.2	52.1	86.6	102.2	113.2	119.8
PGBIG	10.6	36.6	49.1	53.0	56.0	58.6	6.3	**28.7**	49.2	60.4	68.9	73.9	7.1	**30.1**	49.2	58.9	66.4	71.2	9.9	**50.9**	86.2	102.3	112.8	118.4
Ours	**9.9**	35.7	**44.1**	**46.2**	**49.8**	**53.6**	**6.1**	29.0	**44.5**	**52.0**	**59.2**	**65.1**	**6.6**	31.4	**47.6**	**55.2**	**62.4**	**68.1**	**9.6**	54.9	**85.7**	**96.2**	**103.6**	**110.9**
Scenarios	Directions						Greeting						Phoning						Posing					
Model	80ms	320ms	560ms	720ms	880ms	1000ms	80ms	320ms	560ms	720ms	880ms	1000ms	80ms	320ms	560ms	720ms	880ms	1000ms	80ms	320ms	560ms	720ms	880ms	1000ms
Zero-Vel	18.8	64.4	91.6	103.8	114.9	121.1	30.8	97.3	130.6	144.8	156.3	160.5	19.9	66.8	96.5	111.0	121.6	127.5	24.7	87.2	132.4	157.9	179.8	195.0
Res. Sup.	21.6	72.1	101.1	114.5	124.5	129.5	31.2	96.3	126.1	138.8	150.3	153.9	21.1	66.0	94.0	107.7	119.1	126.4	29.3	98.3	140.3	159.8	173.2	183.2
convSeq2Seq	13.5	57.6	86.6	99.8	109.9	115.8	22.0	82.0	116.9	130.7	142.7	147.3	13.5	49.9	77.1	92.1	105.5	114.0	16.9	75.7	122.5	148.8	171.8	187.4
LTD	8.8	46.5	74.2	88.1	99.4	105.5	16.2	68.7	104.8	119.7	132.1	136.8	9.8	40.8	68.8	83.6	96.8	105.1	12.2	63.1	110.2	137.8	160.8	174.8
HRI	7.4	44.5	73.9	88.2	100.1	106.5	13.7	63.8	101.9	118.4	132.7	138.8	8.6	39.0	67.4	82.9	96.5	105.0	10.2	58.5	107.6	136.8	161.4	178.2
PGBIG	7.2	**43.5**	73.1	88.8	100.5	106.1	13.4	**63.1**	100.4	117.7	130.5	136.1	8.4	**38.3**	66.3	82.0	95.4	103.3	9.8	**56.5**	101.5	127.8	149.9	165.3
Ours	**7.0**	46.9	**70.6**	**79.8**	**90.7**	**100.3**	**13.0**	68.8	**98.2**	**106.2**	**116.4**	**126.1**	**8.0**	39.6	**65.1**	**77.3**	**88.8**	**98.0**	**9.0**	59.7	**99.5**	**120.3**	**138.5**	**154.1**
Scenarios	Purchases						Sitting						Sitting Down						Taking Photo					
Model	80ms	320ms	560ms	720ms	880ms	1000ms	80ms	320ms	560ms	720ms	880ms	1000ms	80ms	320ms	560ms	720ms	880ms	1000ms	80ms	320ms	560ms	720ms	880ms	1000ms
Zero-Vel	27.0	80.6	112.1	127.2	139.7	148.0	17.0	56.0	85.2	101.0	114.4	122.7	24.5	74.8	111.0	129.6	144.4	155.1	17.0	57.2	88.4	105.2	118.3	127.2
Res. Sup.	28.7	86.9	122.1	137.2	148.0	154.0	23.8	78.0	113.7	130.5	144.4	152.6	31.7	96.7	138.8	159.0	176.1	187.4	21.9	74.0	110.6	128.9	143.7	153.9
convSeq2Seq	20.3	76.5	111.3	129.1	143.1	151.5	13.5	52.0	82.4	98.8	112.4	120.7	20.7	70.4	106.5	125.1	139.8	150.3	12.7	52.1	84.4	102.4	117.7	128.1
LTD	15.2	64.9	99.2	114.9	127.1	134.9	10.4	46.6	79.2	96.2	110.3	118.7	17.1	63.6	100.2	118.2	133.1	143.8	9.6	43.3	75.3	93.5	108.4	118.8
HRI	13.0	60.4	95.6	110.9	125.0	134.2	9.3	44.3	76.4	93.1	107.0	115.9	14.9	59.1	97.0	116.1	132.1	143.6	8.3	40.7	72.1	90.4	105.5	115.9
PGBIG	12.9	**60.1**	95.6	111.1	123.1	130.6	9.0	**42.5**	74.7	91.3	105.2	114.0	14.5	**58.0**	95.7	114.9	130.1	140.8	**8.1**	**40.1**	72.0	90.2	105.2	115.4
Ours	**12.1**	60.9	**88.9**	**100.0**	**112.3**	**123.3**	**8.7**	43.8	**71.3**	**85.2**	**98.5**	**108.1**	**14.1**	61.3	**94.2**	**110.3**	**124.6**	**135.7**	8.2	42.6	**70.5**	**84.8**	**96.5**	**106.9**
Scenarios	Waiting						Walking Dog						Walking Together						Average					
Model	80ms	320ms	560ms	720ms	880ms	1000ms	80ms	320ms	560ms	720ms	880ms	1000ms	80ms	320ms	560ms	720ms	880ms	1000ms	80ms	320ms	560ms	720ms	880ms	1000ms
Zero-Vel	21.8	72.4	104.6	117.1	125.8	130.3	37.0	99.6	126.7	140.9	154.9	160.8	26.5	85.9	116.5	121.8	123.1	122.7	23.8	76.0	107.4	121.6	131.6	136.6
Res. Sup.	23.8	75.8	105.4	117.3	128.1	135.4	36.4	99.1	128.7	141.1	155.3	164.5	20.4	59.4	80.2	87.3	92.8	98.2	25.0	77.0	106.3	119.4	130.0	136.6
convSeq2Seq	14.6	58.1	87.3	100.3	110.7	117.7	27.7	90.7	122.4	133.8	151.1	162.4	15.3	53.1	72.0	77.7	82.9	87.4	16.6	61.4	90.7	104.7	116.7	124.2
LTD	10.4	47.9	77.2	90.6	101.1	108.3	22.8	77.2	107.8	120.3	136.3	146.4	10.3	39.4	56.0	60.3	63.1	65.7	12.2	50.7	79.6	93.6	105.2	112.4
HRI	8.7	43.4	74.5	89.0	100.3	108.2	20.1	73.3	108.2	120.6	135.9	146.9	8.9	35.1	52.7	57.8	62.0	64.9	10.4	47.1	77.3	91.8	104.1	112.1
PGBIG	8.4	**42.4**	**71.0**	**84.6**	95.6	103.2	19.9	**72.8**	105.5	119.4	135.5	146.1	8.8	**35.4**	54.4	61.0	64.8	67.4	10.3	**46.6**	76.3	90.9	102.6	110.0
Ours	**7.9**	46.2	74.1	84.8	**93.4**	**101.4**	**18.9**	74.7	**101.9**	**111.5**	**126.3**	**139.6**	**8.5**	36.0	**48.5**	**49.9**	**53.3**	**57.9**	**9.9**	48.8	**73.7**	**84.0**	**94.3**	**103.3**

Table D.3: Comparison with deterministic models on Human3.6M [140] Setting-B in MPJPE (mm) at different prediction horizons in different actions. The best results are highlighted in bold, and the second-best ones are marked with underscores.

Bibliography

[1] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," in *Computer graphics forum*, Wiley Online Library, vol. 26, 2007, pp. 655–664.

[2] Y. Luo, P. Cai, A. Bera, D. Hsu, W. S. Lee, and D. Manocha, "Porca: modeling and planning for autonomous driving among many pedestrians," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3418–3425, 2018.

[3] C. Flores, P. Merdrignac, R. de Charette, F. Navas, V. Milanés, and F. Nashashibi, "A cooperative car-following/emergency braking system with prediction-based pedestrian avoidance capabilities," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1837–1846, 2018.

[4] P. Trautman and A. Krause, "Unfreezing the robot: navigation in dense, interacting crowds," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2010, pp. 797–803.

[5] M. Bennewitz, W. Burgard, and S. Thrun, "Learning motion patterns of persons for mobile service robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, vol. 4, 2002.

[6] W. H. Organization, "Global status report on road safety 2018," World Health Organization, Tech. Rep., 2018. [Online]. Available: https://www.who.int/publications/i/item/9789241565684.

[7] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10 684–10 695.

[8] C. Saharia, W. Chan, S. Saxena, *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 36 479–36 494, 2022.

[9] OpenAI, *Gpt-4 technical report*, 2023. arXiv: 2303.08774 [cs.CL].

[10] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine, "One-shot visual imitation learning via meta-learning," in *Proceedings of the 1st Annual Conference on Robot Learning*, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., ser. Proceedings of Machine Learning Research, vol. 78, PMLR, 2017, pp. 357–368.

[11] A. Bubic, D. Y. Von Cramon, and R. Schubotz, "Prediction, cognition and the brain," *Frontiers in Human Neuroscience*, vol. 4, p. 25, 2010, ISSN: 1662-5161. DOI: 10.3389/fnhum.2010.00025. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnhum.2010.00025.

[12] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft coco: common objects in context," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2014, pp. 740–755.

[13] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, "Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior," in *ICCVW*, 2017, pp. 206–213.

[14] P. Kothari, S. Kreiss, and A. Alahi, "Human trajectory forecasting in crowds: a deep learning perspective," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[15] M. Liang, B. Yang, R. Hu, *et al.*, "Learning lane graph representations for motion forecasting," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2020.

[16] S. Saadatnejad, M. Bahari, P. Khorsandi, M. Saneian, S.-M. Moosavi-Dezfooli, and A. Alahi, "Are socially-aware trajectory prediction models really socially-aware?" *Transportation Research Part C: Emerging Technologies*, 2022.

[17] M. Bahari, S. Saadatnejad, A. Rahimi, M. Shaverdikondori, S.-M. Moosavi-Dezfooli, and A. Alahi, "Vehicle trajectory prediction works, but not everywhere," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[18] S. Saadatnejad, Y. Gao, K. Messaoud, and A. Alahi, *Social-transmotion: promptable human trajectory prediction*, 2023.

[19] S. Saadatnejad, M. Mirmohammadi, M. Daghyani, *et al.*, *Toward reliable human pose forecasting with uncertainty*, 2023. arXiv: 2304.06707.

[20] S. Saadatnejad, A. Rasekh, M. Mofayezi, *et al.*, "A generic diffusion-based approach for 3d human pose prediction in the wild," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[21] S. Saadatnejad, S. Li, T. Mordan, and A. Alahi, "A shared representation for photorealistic driving simulators," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[22] M. Bahari, I. Nejjar, and A. Alahi, "Injecting knowledge in data-driven vehicle trajectory predictors," *Transportation Research Part C: Emerging Technologies*, vol. 128, p. 103 010, 2021.

[23] S. Seer, N. Brändle, and C. Ratti, "Kinects and human kinetics: a new approach for studying pedestrian behavior," *Transportation Research Part C: Emerging Technologies*, vol. 48, 2014, ISSN: 0968-090X.

[24] T. Robin, G. Antonini, M. Bierlaire, and J. Cruz, "Specification, estimation and validation of a pedestrian walking behavior model," *Transportation Research Part B: Methodological*, vol. 43, no. 1, 2009.

[25] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: human trajectory prediction in crowded spaces," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[26] V. J. Blue and J. L. Adler, "Cellular automata microsimulation for modeling bi-directional pedestrian walkways," *Transportation Research Part B: Methodological*, vol. 35, no. 3, 2001.

[27] D. C. Duives, W. Daamen, and S. P. Hoogendoorn, "State-of-the-art crowd motion simulation models," *Transportation research part C: emerging technologies*, vol. 37, pp. 193–209, 2013.

[28] N. Nikhil and B. Tran Morris, "Convolutional neural network for trajectory prediction," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018.

[29] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[30] A. Vemula, K. Muelling, and J. Oh, "Social attention: modeling attention in human crowds," in *IEEE international Conference on Robotics and Automation (ICRA)*, IEEE, 2018.

[31] V. Kosaraju, A. Sadeghian, R. Martin-Martin, I. Reid, H. Rezatofighi, and S. Savarese, "Social-bigat: multimodal trajectory forecasting using bicycle-gan and graph attention networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[32] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "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 (CVPR)*, 2020, pp. 14 424–14 432.

[33] G. Ortiz-Jimenez, A. Modas, S.-M. Moosavi, and P. Frossard, "Hold me tight! influence of discriminative features on deep network boundaries," in *Advances in Neural Information Processing Systems (NeurIPS)*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020.

[34] C. Szegedy, W. Zaremba, I. Sutskever, *et al.*, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[35] W. Deng, L. Bertoni, S. Kreiss, and A. Alahi, "Joint human pose estimation and stereo 3d localization," in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020.

[36] L. Bertoni, S. Kreiss, T. Mordan, and A. Alahi, "Monstereo: when monocular and stereo meet at the tail of 3d human localization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[37] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, May 1998. DOI: 10.1103/PhysRevE.51.4282.

[38] G. Antonini, M. Bierlaire, and M. Weber, "Discrete choice models of pedestrian walking behavior," *Transportation Research Part B: Methodological*, vol. 40, no. 8, pp. 667–687, 2006.

[39] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz, "Simulation of pedestrian dynamics using a two-dimensional cellular automaton," *Physica A: Statistical Mechanics and its Applications*, vol. 295, no. 3-4, pp. 507–525, 2001.

[40] G. Vizzari, L. Manenti, K. Ohtsuka, and K. Shimura, "An agent-based pedestrian and group dynamics model applied to experimental and real-world scenarios," *Journal of Intelligent Transportation Systems*, vol. 19, no. 1, pp. 32–45, 2015.

[41] A. Alahi, V. Ramanathan, K. Goel, *et al.*, "Learning to predict human behavior in crowded scenes," in *Group and Crowd Behavior for Computer Vision*, Elsevier, 2017.

[42] F. Bartoli, G. Lisanti, L. Ballan, and A. Del Bimbo, "Context-aware trajectory prediction," in *International Conference on Pattern Recognition (ICPR)*, IEEE, 2018.

[43] B. Ivanovic and M. Pavone, "The trajectron: probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[44] J. Li, H. Ma, Z. Zhang, and M. Tomizuka, "Social-wagdat: interaction-aware trajectory prediction via wasserstein graph double-attention network," *arXiv preprint arXiv:2002.06241*, 2020.

[45] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, "Spatio-temporal graph transformer networks for pedestrian trajectory prediction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2020, pp. 507–523.

[46] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "Sophie: an attentive gan for predicting paths compliant to social and physical constraints," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1349–1358.

[47] J. Amirian, J.-B. Hayet, and J. Pettre, "Social ways: learning multi-modal distributions of pedestrian trajectories with gans," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019.

[48] W. Zeng, W. Luo, S. Suo, *et al.*, "End-to-end interpretable neural motion planner," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8660–8669.

[49] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 285–292.

[50] T. van der Heiden, N. S. Nagaraja, C. Weiss, and E. Gavves, "Safecritic: collision-aware trajectory prediction," in *British Machine Vision Conference Workshop*, 2019.

[51] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[52] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2574–2582.

[53] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

[54] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li, "Adversarial attacks on deep-learning models in natural language processing: a survey," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 11, no. 3, pp. 1–41, 2020.

[55] M. Behjati, S.-M. Moosavi-Dezfooli, M. S. Baghshah, and P. Frossard, "Universal adversarial attacks on text classifiers," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019.

[56] V. Ballet, X. Renard, J. Aigrain, T. Laugel, P. Frossard, and M. Detyniecki, "Imperceptible adversarial attacks on tabular data," *arXiv preprint arXiv:1911.03274*, 2019.

[57] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Adversarial attacks on deep neural networks for time series classification," in *2019 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2019, pp. 1–8.

[58] F. Karim, S. Majumdar, and H. Darabi, "Adversarial attacks on time series," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[59] J. Liu, N. Akhtar, and A. Mian, "Adversarial attack on skeleton-based human action recognition," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[60] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[61] K. Mangalam, H. Girase, S. Agarwal, *et al.*, "It is not the journey but the destination: endpoint conditioned trajectory prediction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2020, pp. 759–776.

[62] S. Pellegrini, A. Ess, and L. V. Gool, "Improving data association by joint modeling of pedestrian trajectories and groupings," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2010, pp. 452–465.

[63] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," *Comput. Graph. Forum*, vol. 26, pp. 655–664, 2007.

[64] T. Chavdarova, P. Baque, S. Bouquet, *et al.*, "Wildtrack: a multi-camera hd dataset for dense unscripted pedestrian detection," *Prooceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5030–5039, 2018.

[65] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: human trajectory understanding in crowded scenes," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2016.

[66] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," *arXiv preprint arXiv: 1805.12152*, 2018.

[67] S. Becker, R. Hug, W. Hübner, and M. Arens, "An evaluation of trajectory prediction approaches and notes on the trajnet benchmark," *arXiv preprint arXiv:1805.07663*, 2018.

[68] I. Karamouzas, B. Skinner, and S. J. Guy, "Universal power law governing pedestrian interactions," *Physical review letters*, vol. 113, no. 23, p. 238 701, 2014.

[69] S. Ettinger, S. Cheng, B. Caine, *et al.*, "Large scale interactive motion forecasting for autonomous driving: the waymo open motion dataset," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[70] M.-F. Chang, J. Lambert, P. Sangkloy, *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[71] H. Caesar, V. Bankiti, A. H. Lang, *et al.*, "Nuscenes: a multimodal dataset for autonomous driving," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[72] S. H. Park, G. Lee, J. Seo, *et al.*, "Diverse and admissible trajectory forecasting through multimodal context understanding," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2020.

[73] S. Casas, C. Gulino, S. Suo, and R. Urtasun, *The importance of prior knowledge in precise multimodal prediction*, 2020. eprint: 2006.02636.

[74] M. Bansal, A. Krizhevsky, and A. Ogale, "Chauffeurnet: learning to drive by imitating the best and synthesizing the worst," *Robotics: Science and Systems (RSS)*, 2019.

[75] M. Niedoba, H. Cui, K. Luo, D. Hegde, F.-C. Chou, and N. Djuric, "Improving movement prediction of traffic actors using off-road loss and bias mitigation," *Advances in Neural Information Processing Systems (NeurIPS) Workshop*, 2019.

[76] L. Mi, H. Zhao, C. Nash, *et al.*, "Hdmapgen: a hierarchical graph generative model of high definition maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[77] *Openstreetmap*, https://www.openstreetmap.org.

[78] P. Coscia, F. Castaldo, F. A. Palmieri, A. Alahi, S. Savarese, and L. Ballan, "Long-term path prediction in urban scenarios using circular distributions," *Image and Vision Computing*, vol. 69, 2018, ISSN: 0262-8856. DOI: https://doi.org/10.1016/j.imavis.2017.11.006.

[79] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: distant future prediction in dynamic scenes with interacting agents," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 336–345.

[80] X. Ren, T. Yang, L. E. Li, A. Alahi, and Q. Chen, "Safety-aware motion prediction with unseen vehicles for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[81] Y. Biktairov, M. Stebelev, I. Rudenko, O. Shliazhko, and B. Yangel, "Prank: motion prediction based on ranking," *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[82] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "Multipath: multiple probabilistic anchor trajectory hypotheses for behavior prediction," *Conference on Robot Learning*, 2019.

[83] J. Hong, B. Sapp, and J. Philbin, "Rules of the road: predicting driving behavior with a convolutional model of semantic interactions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[84] S. Casas, W. Luo, and R. Urtasun, "Intentnet: learning to predict intention from raw sensor data," in *Conference on Robot Learning*, PMLR, 2018.

[85] A. Sadeghian, F. Legros, M. Voisin, R. Vesel, A. Alahi, and S. Savarese, "Car-net: clairvoyant attentive recurrent network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[86] J. Gao, C. Sun, H. Zhao, *et al.*, "Vectornet: encoding hd maps and agent dynamics from vectorized representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[87] H. Song, D. Luan, W. Ding, M. Y. Wang, and Q. Chen, "Learning to predict vehicle trajectories with model-based planning," in *Conference on Robot Learning*, 2021.

[88] M. Bahari, V. Zehtab, S. Khorasani, S. Ayramlou, S. Saadatnejad, and A. Alahi, "Svg-net: an svg-based trajectory prediction model," *arXiv preprint arXiv:2110.03706*, 2021.

[89] N. Rhinehart, K. M. Kitani, and P. Vernaza, "R2p2: a reparameterized pushforward policy for diverse, precise generative path forecasting," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[90] F. Indaheng, E. Kim, K. Viswanadha, *et al.*, "A scenario-based platform for testing autonomous vehicle behavior prediction models in simulation," *arXiv preprint arXiv:2110.14870*, 2021.

[91] Y. Abeysirigoonawardena, F. Shkurti, and G. Dudek, "Generating adversarial driving scenarios in high-fidelity simulators," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[92] A. Wachi, "Failure-scenario maker for rule-based agent using multi-agent adversarial reinforcement learning and its application to autonomous driving," *Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

[93] M. Althoff and S. Lutz, "Automatic generation of safety-critical test scenarios for collision avoidance of road vehicles," in *IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[94] M. Klischat and M. Althoff, "Generating critical test scenarios for automated vehicles with evolutionary algorithms," in *IEEE Intelligent Vehicles Symposium (IV)*, 2019. DOI: 10.1109/IVS.2019.8814230.

[95] A. Corso, R. J. Moss, M. Koren, R. Lee, and M. J. Kochenderfer, "A survey of algorithms for black-box safety validation," *arXiv preprint arXiv:2005.02979*, 2020.

[96] Y. Cao, C. Xiao, D. Yang, *et al.*, "Adversarial objects against lidar-based autonomous driving systems," *arXiv preprint arXiv:1907.05418*, 2019.

[97] J. Sun, Y. Cao, Q. A. Chen, and Z. M. Mao, "Towards robust lidar-based perception in autonomous driving: general black-box adversarial sensor attack and countermeasures," in *29th USENIX Security Symposium*, 2020.

[98] J. Tu, M. Ren, S. Manivasagam, *et al.*, "Physically realizable adversarial examples for lidar object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[99] J. Wang, A. Pun, J. Tu, *et al.*, "Advsim: generating safety-critical scenarios for self-driving vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[100] Y. Tian, K. Pei, S. Jana, and B. Ray, "Deeptest: automated testing of deep-neural-network-driven autonomous cars," in *Proceedings of the 40th international conference on software engineering*, 2018, pp. 303–314.

[101] M. Zhang, Y. Zhang, L. Zhang, C. Liu, and S. Khurshid, "Deeproad: gan-based metamorphic testing and input validation framework for autonomous driving systems," in *IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2018.

[102] H. Machiraju and V. N. Balasubramanian, "A little fog for a large turn," in *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020.

[103] Z. Kong, J. Guo, A. Li, and C. Liu, "Physgan: generating physical-world-resilient adversarial examples for autonomous driving," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[104] H. Zhou, W. Li, Z. Kong, *et al.*, "Deepbillboard: systematic physical-world testing of autonomous driving systems," in *Proceedings of the ACM/IEEE International Conference on Software Engineering*, 2020, pp. 347–358.

[105] T. Sato, J. Shen, N. Wang, Y. Jia, X. Lin, and Q. A. Chen, "Dirty road can attack: security of deep learning based automated lane centering under physical-world attack," in *Proceedings of the 29th USENIX Security Symposium (USENIX Security'21)*, 2021.

[106] A. Boloor, K. Garimella, X. He, C. Gill, Y. Vorobeychik, and X. Zhang, "Attacking vision-based perception in end-to-end autonomous driving models," *Journal of Systems Architecture*, vol. 110, 2020, ISSN: 1383-7621. DOI: https://doi.org/10.1016/j.sysarc.2020.101766.

[107] H. David, R. Resnick, and J. Walker, *Fundamentals of physics.* Wiley, 1997.

[108] B. Ru, A. D. Cobb, A. Blaas, and Y. Gal, "Bayesopt adversarial attack," *International Conference on Learning Representations (ICLR)*, 2020.

[109] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Gaussian process optimization in the bandit setting: no regret and experimental design," *International Conference on Machine Learning (ICML)*, 2010.

[110] M. Alzantot, Y. Sharma, S. Chakraborty, H. Zhang, C.-J. Hsieh, and M. B. Srivastava, "Genattack: practical black-box attacks with gradient-free optimization.," *GECCO*, 2019.

[111] K.-F. Man, W. K.-S. Tang, and S. Kwong, "Genetic algorithms: concepts and applications [in engineering design]," *IEEE Trans. Ind. Electron.*, 1996.

[112] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization.," *Advances in Neural Information Processing Systems(NeurIPS)*, 2011.

[113] S. Khandelwal, W. Qi, J. Singh, A. Hartnett, and D. Ramanan, "What-if motion prediction for autonomous driving," *arXiv preprint arXiv:2008.10587*, 2020.

[114] "Argoai challenge," *CVPR Workshop on Autonomous Driving*, 2020. [Online]. Available: %5Curl%7Bhttps://www.youtube.com/watch?v=Vcbj%5C_peZT4Q%7D.

[115] J. Ziegler, P. Bender, M. Schreiber, *et al.*, "Making bertha drive—an autonomous journey on a historic route," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, 2014.

[116] P. J. Blau, *Friction Science and Technology: From Concepts to Applications*. CRC Press, 2005.

[117] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2006.

[118] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, 1995.

[119] G. Best and R. Fitch, "Bayesian intention inference for trajectory prediction with an unknown goal destination," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2015, pp. 5817–5823.

[120] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: human trajectory prediction in crowded spaces," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 961–971.

[121] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2255–2264.

[122] F. Giuliari, I. Hasan, M. Cristani, and F. Galasso, "Transformer networks for trajectory forecasting," in *International Conference on Pattern Recognition (ICPR)*, IEEE, 2021, pp. 10 335–10 342.

[123] A. Monti, A. Bertugli, S. Calderara, and R. Cucchiara, "Dag-net: double attentive graph neural network for trajectory forecasting," in *International Conference on Pattern Recognition (ICPR)*, IEEE, 2021, pp. 2551–2558.

[124] J. Sun, Y. Li, L. Chai, H.-S. Fang, Y.-L. Li, and C. Lu, "Human trajectory prediction with momentary observation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 6467–6476.

[125] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng, "Sr-lstm: state refinement for lstm towards pedestrian trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 085–12 094.

[126] K. Mangalam, Y. An, H. Girase, and J. Malik, "From goals, waypoints & paths to long term human trajectory forecasting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 15 233–15 242.

[127] A. Bhattacharyya, D. O. Reino, M. Fritz, and B. Schiele, "Euro-pvi: pedestrian vehicle interactions in dense urban centers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 6408–6417.

[128] C. Zhang and C. Berger, "Learning the pedestrian-vehicle interaction for pedestrian trajectory prediction," in *International Conference on Control, Automation and Robotics (ICCAR)*, IEEE, 2022, pp. 230–236.

[129] Y. Hu, S. Chen, Y. Zhang, and X. Gu, "Collaborative motion prediction via neural motion message passing," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR*, 2020, pp. 6319–6328.

[130] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang, "Stgat: modeling spatial-temporal interactions for human trajectory prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6272–6281.

[131] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: dynamically-feasible trajectory forecasting with heterogeneous data," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2020, pp. 683–700.

[132] T. Gu, G. Chen, J. Li, *et al.*, "Stochastic trajectory prediction via motion indeterminacy diffusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 17 113–17 122.

[133] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 6000–6010.

[134] L. Li, M. Pagnucco, and Y. Song, "Graph-based spatial transformer with memory replay for multi-future pedestrian trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 2231–2241.

[135] Y. Yuan, X. Weng, Y. Ou, and K. M. Kitani, "Agentformer: agent-aware transformers for socio-temporal multi-agent forecasting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9813–9823.

[136] R. Girgis, F. Golemo, F. Codevilla, *et al.*, "Latent variable sequential set transformers for joint multi-agent motion prediction," in *International Conference on Learning Representations (ICLR)*, 2022. [Online]. Available: https://openreview.net/forum?id=Dup_dDqkZC5.

[137] C. Xu, R. T. Tan, Y. Tan, *et al.*, "Eqmotion: equivariant multi-agent motion prediction with invariant interaction reasoning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 1410–1420.

[138] M. Fabbri, F. Lanzi, S. Calderara, A. Palazzi, R. Vezzani, and R. Cucchiara, "Learning to detect and track visible and occluded body joints in a virtual world," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 430–446.

[139] R. Martin-Martin, M. Patel, H. Rezatofighi, *et al.*, "Jrdb: a dataset and benchmark of egocentric robot visual perception of humans in built environments," *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[140] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, 2014.

[141] S. Bouhsain, S. Saadatnejad, and A. Alahi, "Pedestrian intention prediction: a multi-task perspective," in *European Association for Research in Transportation (hEART)*, 2020.

[142] S. Saadatnejad, Y. Z. Ju, and A. Alahi, "Pedestrian 3d bounding box prediction," in *European Association for Research in Transportation (hEART)*, 2022.

[143] J. Liang, L. Jiang, J. C. Niebles, A. G. Hauptmann, and L. Fei-Fei, "Peeking into the future: predicting future person activities and locations in videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5725–5734.

[144] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "Openpose: realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[145] T. Yagi, K. Mangalam, R. Yonetani, and Y. Sato, "Future person localization in first-person videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7593–7602.

[146] K. Chen, X. Song, and X. Ren, "Pedestrian trajectory prediction in heterogeneous traffic using pose keypoints-based convolutional encoder-decoder network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 1764–1775, 2020.

[147] B. Wandt, M. Rudolph, P. Zell, H. Rhodin, and B. Rosenhahn, "Canonpose: self-supervised monocular 3d human pose estimation in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 13 294–13 304.

[148] B. Parsaeifard, S. Saadatnejad, Y. Liu, T. Mordan, and A. Alahi, "Learning decoupled representations for human pose forecasting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) workshop*, 2021, pp. 2294–2303.

[149] V. Adeli, E. Adeli, I. Reid, J. C. Niebles, and H. Rezatofighi, "Socially and contextually aware human motion and pose forecasting," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6033–6040, 2020.

[150] C. Wang, Y. Wang, Z. Huang, and Z. Chen, "Simple baseline for single human motion forecasting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 2260–2265.

[151] N. D. Reddy, L. Guigues, L. Pishchulin, J. Eledath, and S. G. Narasimhan, "Tessetrack: end-to-end learnable multi-person articulated 3d pose tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 15 190–15 200.

[152] V. Kress, F. Jeske, S. Zernetsch, K. Doll, and B. Sick, "Pose and semantic map based probabilistic forecast of vulnerable road users trajectories," *IEEE Transactions on Intelligent Vehicles*, 2022.

[153] I. Hasan, F. Setti, T. Tsesmelis, *et al.*, "Forecasting people trajectories and head poses by jointly reasoning on tracklets and vislets," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 4, pp. 1267–1278, 2019.

[154] X. Du, R. Vasudevan, and M. Johnson-Roberson, "Bio-lstm: a biomechanically inspired recurrent neural network for 3-d pedestrian pose and gait prediction," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1501–1508, 2019. DOI: 10.1109/LRA.2019.2895266.

[155] L. Vianello, J.-B. Mouret, E. Dalin, A. Aubry, and S. Ivaldi, "Human posture prediction during physical human-robot interaction," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 6046–6053, 2021.

[156] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: crowd-aware robot navigation with attention-based deep reinforcement learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[157] F. B. Wagner, J.-B. Mignardot, C. G. Le Goff-Mignardot, *et al.*, "Targeted neurotechnology restores walking in humans with spinal cord injury," *Nature*, vol. 563, no. 7729, pp. 65–71, 2018.

[158] W. Mao, M. Liu, and M. Salzmann, "History repeats itself: human motion prediction via motion attention," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2020, pp. 474–489.

[159] T. Sofianos, A. Sampieri, L. Franco, and F. Galasso, "Space-time-separable graph convolutional network for pose forecasting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 11 209–11 218.

[160] T. Ma, Y. Nie, C. Long, Q. Zhang, and G. Li, "Progressively generating better initial guesses towards next stages for high-quality human motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 6437–6446.

[161] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[162] L. Bertoni, S. Kreiss, and A. Alahi, "MonoLoco: monocular 3D pedestrian localization and uncertainty estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[163] J. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax Weiss, and B. Lakshminarayanan, "Simple and principled uncertainty estimation with deterministic deep learning via distance awareness," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 7498–7512, 2020.

[164] J. Van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal, "Uncertainty estimation using a single deep deterministic neural network," in *International Conference on Machine Learning (ICML)*, PMLR, 2020, pp. 9690–9700.

[165] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: representing model uncertainty in deep learning," in *International Conference on Machine Learning (ICML)*, PMLR, 2016, pp. 1050–1059.

[166] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "Amass: archive of motion capture as surface shapes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. [Online]. Available: https://amass.is.tue.mpg.de.

[167] T. von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll, "Recovering accurate 3d human pose in the wild using imus and a moving camera," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 601–617.

[168] M. Hassan, D. Ceylan, R. Villegas, *et al.*, "Stochastic scene-aware motion prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 11 374–11 384.

[169] Y. Cai, Y. Wang, Y. Zhu, *et al.*, "A unified 3d human motion synthesis model via conditional variational auto-encoder," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 11 645–11 655.

[170] C. Li, Z. Zhang, W. S. Lee, and G. H. Lee, "Convolutional sequence to sequence model for human dynamics," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5226–5234.

[171] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: deep learning on spatio-temporal graphs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5308–5317.

[172] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2891–2900.

[173] W. Mao, M. Liu, M. Salzmann, and H. Li, "Learning trajectory dependencies for human motion prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[174] C. Zhong, L. Hu, Z. Zhang, Y. Ye, and S. Xia, "Spatio-temporal gating-adjacency gcn for human motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 6447–6456.

[175] Y. Yuan and K. Kitani, "Dlow: diversifying latent flows for diverse human motion prediction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[176] S. Aliakbarian, F. S. Saleh, M. Salzmann, L. Petersson, and S. Gould, "A stochastic conditioning scheme for diverse human motion prediction," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 5223–5232.

[177] S. Aliakbarian, F. Saleh, L. Petersson, S. Gould, and M. Salzmann, "Contextually plausible and diverse 3d human motion prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 11 333–11 342.

[178] T. Salzmann, M. Pavone, and M. Ryll, "Motron: multimodal probabilistic human motion forecasting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 6457–6466.

[179] H. Ma, J. Li, R. Hosseini, M. Tomizuka, and C. Choi, "Multi-objective diverse human motion prediction with knowledge distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 8161–8171.

[180] W. Mao, M. Liu, and M. Salzmann, "Generating smooth pose sequences for diverse human motion prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 13 309–13 318.

[181] P. Nikdel, M. Mahdavian, and M. Chen, "Dmmgan: diverse multi motion prediction of 3d human joints using attention-based generative adversarial network," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 9938–9944. DOI: 10.1109/ICRA48891.2023.10160401.

[182] L. Dang, Y. Nie, C. Long, Q. Zhang, and G. Li, "Msr-gcn: multi-scale residual graph convolution networks for human motion prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 11 467–11 476.

[183] R. McAllister, Y. Gal, A. Kendall, *et al.*, "Concrete problems for autonomous vehicle safety: advantages of bayesian deep learning," International Joint Conferences on Artificial Intelligence, Inc., 2017.

[184] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv preprint arXiv:1511.02680*, 2015.

[185] Y. Xiao and W. Y. Wang, "Quantifying uncertainties in natural language processing tasks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, 2019, pp. 7322–7329.

[186] J. Kundu, S. Seth, P. YM, V. Jampani, A. Chakraborty, and R. Babu, "Uncertainty-aware adaptation for self-supervised 3d human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 20 416–20 427. DOI: 10.1109/CVPR52688.2022.01980.

[187] T. M. Iversen, A. G. Buch, and D. Kraft, "Prediction of icp pose uncertainties using monte carlo simulation with synthetic depth images," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4640–4647. DOI: 10.1109/IROS.2017.8206335.

[188] S. Prokudin, P. Gehler, and S. Nowozin, "Deep directional statistics: pose estimation with uncertainty quantification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[189] X. Tang, K. Yang, H. Wang, *et al.*, "Prediction-uncertainty-aware decision-making for autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, pp. 1–15, 2022. DOI: 10.1109/TIV.2022.3188662.

[190] B. Ivanovic, Y. Lin, S. Shrivastava, P. Chakravarty, and M. Pavone, "Propagating state uncertainty through trajectory forecasting," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2351–2358. DOI: 10.1109/ICRA46639.2022.9811776.

[191] X. Huang, S. G. McGill, B. C. Williams, L. Fletcher, and G. Rosman, "Uncertainty-aware driver trajectory prediction at urban intersections," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 9718–9724. DOI: 10.1109/ICRA.2019.8794282.

[192] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.

[193] A. Graves, "Practical variational inference for neural networks," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 24, 2011.

[194] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *International Conference on Machine Learning (ICML)*, PMLR, 2015, pp. 1613–1622.

[195] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[196] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on Machine Learning (ICML)*, PMLR, 2017, pp. 1321–1330.

[197] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[198] J. H. Ricketts and G. A. Head, "A five-parameter logistic equation for investigating asymmetry of curvature in baroreflex studies," *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, vol. 277, no. 2, R441–R454, 1999.

[199] Y. Wang, Z. Shi, X. Guo, X. Liu, E. Zhu, and J. Yin, "Deep embedding for determining the number of clusters," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[200] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[201] J. Hou and M. Pelillo, "A new density kernel in density peak based clustering," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, IEEE, 2016, pp. 468–473.

[202] J. Wang, Y. Zhang, and X. Lan, "Automatic cluster number selection by finding density peaks," in *IEEE International Conference on Computer and Communications (ICCC)*, IEEE, 2016, pp. 13–18.

[203] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International Conference on Machine Learning (ICML)*, PMLR, 2016, pp. 478–487.

[204] S. Xu, Y.-X. Wang, and L.-Y. Gui, "Diverse human motion prediction guided by multi-level spatial-temporal anchors," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[205] Y. Tashiro, J. Song, Y. Song, and S. Ermon, "Csdi: conditional score-based diffusion models for probabilistic time series imputation," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, pp. 24 804–24 816, 2021.

[206] R. El-Yaniv *et al.*, "On the foundations of noise-free selective classification.," *Journal of Machine Learning Research*, vol. 11, no. 5, 2010.

[207] J. Ren, P. J. Liu, E. Fertig, *et al.*, "Likelihood ratios for out-of-distribution detection," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[208] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 6840–6851, 2020.

[209] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, pp. 8780–8794, 2021.

[210] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "Repaint: inpainting using denoising diffusion probabilistic models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11 461–11 471.

[211] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10 684–10 695.

[212] Q. Cui and H. Sun, "Towards accurate 3d human motion prediction from incomplete observations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 4801–4810.

[213] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *International Conference on Machine Learning (ICML)*, PMLR, 2021, pp. 8162–8171.

[214] L. Sigal, A. Balan, and M. J. Black, "HumanEva: synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *International Journal of Computer Vision*, vol. 87, no. 1, pp. 4–27, Mar. 2010.

[215] J. Walker, K. Marino, A. Gupta, and M. Hebert, "The pose knows: video forecasting by generating pose futures," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017, pp. 3332–3341.

[216] X. Yan, A. Rastogi, R. Villegas, *et al.*, "Mt-vae: learning motion transformations to generate multimodal human dynamics," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 265–281.

[217] E. Barsoum, J. R. Kender, and Z. Liu, "Hp-gan: probabilistic 3d human motion prediction via gan," *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1499–149 909, 2018.

[218] A. Bhattacharyya, B. Schiele, and M. Fritz, "Accurate and diverse sampling of sequences based on a "best of many" sample objective," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8485–8493, 2018.

[219] N. Dilokthanakul, P. A. M. Mediano, M. Garnelo, *et al.*, "Deep unsupervised clustering with gaussian mixture variational autoencoders," *CoRR*, vol. abs/1611.02648, 2016. arXiv: 1611.02648.

[220] S. Gurumurthy, R. K. Sarvadevabhatla, and R. V. Babu, "Deligan: generative adversarial networks for diverse and limited data," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4941–4949, 2017.

[221] Y. Yuan and K. Kitani, "Diverse trajectory forecasting with determinantal point processes," *arXiv preprint arXiv:1907.04967*, 2019.

[222] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[223] Q. Cui, H. Sun, and F. Yang, "Learning dynamic relationships for 3d human motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 6519–6527.

[224] M. Li, S. Chen, Y. Zhao, Y. Zhang, Y. Wang, and Q. Tian, "Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 214–223.

[225] A. Hernandez, J. Gall, and F. Moreno-Noguer, "Human motion prediction via spatio-temporal inpainting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 7134–7143.

[226] N. Kalra and S. M. Paddock, "Driving to safety: how many miles of driving would it take to demonstrate autonomous vehicle reliability?" *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182–193, 2016.

[227] Y. F. Payalan and M. A. Guvensan, "Towards next-generation vehicles featuring the vehicle intelligence," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 30–47, 2020.

[228] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[229] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[230] Z. Zheng, X. Yang, Z. Yu, L. Zheng, Y. Yang, and J. Kautz, "Joint discriminative and generative learning for person re-identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[231] J. Liu, B. Ni, Y. Yan, P. Zhou, S. Cheng, and J. Hu, "Pose transferrable person re-identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[232] S. Beery, Y. Liu, D. Morris, *et al.*, "Synthetic examples improve generalization for rare classes," *CoRR*, vol. abs/1904.05916, 2019.

[233] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: an open urban driving simulator," in *Conference on robot learning*, PMLR, 2017.

[234] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[235] A. Amini, I. Gilitschenski, J. Phillips, *et al.*, "Learning robust control policies for end-to-end autonomous driving from data-driven simulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1143–1150, 2020.

[236] Z. Yang, Y. Chai, D. Anguelov, *et al.*, "Surfelgan: synthesizing realistic sensor data for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[237] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[238] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[239] A. Siarohin, E. Sangineto, S. Lathuilière, and N. Sebe, "Deformable GANs for pose-based human image generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[240] M. Tancik, V. Casser, X. Yan, *et al.*, "Block-nerf: scalable large scene neural view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 8248–8258.

[241] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2014, pp. 2672–2680.

[242] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations (ICLR)*, 2014.

[243] P. Esser, E. Sutter, and B. Ommer, "A variational U-Net for conditional appearance and shape generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[244] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4401–4410.

[245] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *International Conference on Learning Representations (ICLR)*, 2019.

[246] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," *International Conference on Learning Representations (ICLR)*, 2018.

[247] Z. Zheng, Y. Wu, X. Han, and J. Shi, "Forkgan: seeing into the rainy night," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[248] E. Romera, L. M. Bergasa, K. Yang, J. M. Alvarez, and R. Barea, "Bridging the day and night domain gap for semantic segmentation," in *Intelligent Vehicles Symposium (IV)*, IEEE, 2019, pp. 1312–1318.

[249] J. Liu, Z. Ma, Y. Qiu, X. Ni, B. Shi, and H. Liu, "Four discriminator cycle-consistent adversarial network for improving railway defective fastener inspection," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2021.

[250] T. R. Shaham, M. Gharbi, R. Zhang, E. Shechtman, and T. Michaeli, "Spatially-adaptive pixelwise networks for fast image translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 14 882–14 891.

[251] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky, "Few-shot adversarial learning of realistic neural talking head models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9459–9468.

[252] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *International Conference on Machine Learning (ICML)*, 2017, pp. 2642–2651.

[253] H. Tang, D. Xu, N. Sebe, Y. Wang, J. J. Corso, and Y. Yan, "Multi-channel attention selection gan with cascaded semantic guidance for cross-view image translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[254] Y. Alami Mejjati, C. Richardt, J. Tompkin, D. Cosker, and K. I. Kim, "Unsupervised attention-guided image-to-image translation," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018, pp. 3693–3703.

[255] A. Cherian and A. Sullivan, "Sem-gan: semantically-consistent image-to-image translation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2019, pp. 1797–1806.

[256] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[257] A. Pumarola, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "Unsupervised person image synthesis in arbitrary poses," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[258] L. Ma, Q. Sun, S. Georgoulis, L. Van Gool, B. Schiele, and M. Fritz, "Disentangled person image generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[259] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. V. Gool, "Pose guided person image generation," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[260] S. Saadatnejad and A. Alahi, "Pedestrian image generation for self-driving cars," in *STRC*, 2019.

[261] H. Dong, X. Liang, K. Gong, H. Lai, J. Zhu, and J. Yin, "Soft-gated warping-GAN for pose-guided person image synthesis," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018, pp. 474–484.

[262] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, "Everybody dance now," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[263] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, *et al.*, "Video-to-video synthesis," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018, pp. 1144–1156.

[264] T.-C. Wang, M.-Y. Liu, A. Tao, G. Liu, J. Kautz, and B. Catanzaro, "Few-shot video-to-video synthesis," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[265] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[266] Y. Liu, P. Kothari, and A. Alahi, "Collaborative sampling in generative adversarial networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[267] P. Li, X. Liang, D. Jia, and E. P. Xing, "Semantic-aware grad-GAN for virtual-to-real urban scene adaption," *British Machine Vision Conference (BMVC)*, 2018.

[268] E. Schonfeld, B. Schiele, and A. Khoreva, "A u-net based discriminator for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8207–8216.

[269] X. Liu, G. Yin, J. Shao, X. Wang, *et al.*, "Learning to predict layout-to-image conditional convolutions for semantic image synthesis," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 570–580.

[270] R. Tyleček and R. Šára, "Spatial pattern templates for recognition of objects with regular structure," in *Proceedings of the German Conference on Pattern Recognition (GCPR)*, 2013.

[271] M. Cordts, M. Omran, S. Ramos, *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[272] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[273] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 472–480.

[274] H.-P. Huang, H.-Y. Tseng, H.-Y. Lee, and J.-B. Huang, "Semantic view synthesis," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Springer, 2020, pp. 592–608.

[275] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: powering robust clothes recognition and retrieval with rich annotations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[276] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[277] W. Nie, B. Guo, Y. Huang, C. Xiao, A. Vahdat, and A. Anandkumar, "Diffusion models for adversarial purification," in *International Conference on Machine Learning (ICML)*, 2022.

[278] Y. Hu, J. Yang, L. Chen, *et al.*, "Planning-oriented autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[279] Z. Liu, P. Su, S. Wu, *et al.*, "Motion prediction using trajectory cues," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 13 299–13 308.

Saeed Saadatnejad |

Address: – Lausanne, Switzerland
(+41) 762226448 • saeed.saadatnejad@epfl.ch
https://saeedsaadatnejad.github.io
linkedin.com/in/saeedsaadatnejad/ Github Scholar

Education

École polytechnique fédérale de Lausanne (EPFL)
PhD, Computer Science, GPA: 5.5/6 *2018–2023*
Thesis: Deep Generative Models for Autonomous Driving: Motion Forecasting to Realistic Image Synthesis
(Doctorate *Award Nominations* at EPFL) Advisor: Prof. Alexandre Alahi

Sharif University of Technology
Master of Science, Computer Science, GPA: 5.75/6 *2015–2018*
Thesis: A Novel ECG Classification Algorithm based on Deep Learning

Sharif University of Technology
Bachelor of Science, Electrical Engineering, GPA: 5.5/6 *2011–2015*

Research Experience

research assistant
VITA EPFL, Lausanne *2018–2023*
Generative models (GANs, VAEs, Diffusion, . . .) for image synthesis and human motion forecasting

computer vision research intern
Disney Research Studios, Zurich *Jul – Sep 2022*
Developing a new generative model for 3D human synthesis

computer vision research intern
Valeo ai, Paris *May – Oct 2021*
Human behavior prediction for autonomous vehicles

Selected Publications

2023: **Saeed Saadatnejad**, A. Rasekh, M. Mofayezi, Y. Medghalchi, S. Rajabzadeh, T. Mordan, and A. Alahi. A generic diffusion-based approach for 3d human pose prediction in the wild. In *International Conference on Robotics and Automation (ICRA)*, 2023.

2023: **Saeed Saadatnejad**, M. Mirmohammadi, M. Daghyani, T. Mordan, and A. Alahi. Toward reliable human pose forecasting with uncertainty. *under review*, 2023.

2023: **Saadatnejad Saeed**, Y. Gao, K. Messaoud, and A. Alahi. Social-transmotion: Promptable human trajectory prediction. *under review*, 2023.

2022: **Saeed Saadatnejad***, M. Bahari*, A. Rahimi, M. Shaverdikondori, S.-M. Moosavi-Dezfooli, and A. Alahi. Vehicle trajectory prediction works, but not everywhere. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

2022: **Saeed Saadatnejad***, M. Bahari*, P. Khorsandi, M. Saneian, S.-M. Moosavi-Dezfooli, and A. Alahi. Are socially-aware trajectory prediction models really socially-aware? *Transportation Research Part C: Emerging Technologies (TR_C)*, 2022.

2021: **Saeed Saadatnejad**, S. Li, T. Mordan, and A. Alahi. A shared representation for photorealistic driving simulators. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

2020: **Saeed Saadatnejad**, S. Bouhsain, and A. Alahi. Pedestrian intention prediction: A multi-task perspective. In *hEART*, 2020.

2019: **Saeed Saadatnejad**, M. Oveisi, and M. Hashemi. Lstm-based ecg classification for continuous monitoring on personal wearable devices. *IEEE Journal of Biomedical and Health Informatics*, 2019.

Honors

2018 – 2023: Awarded highly competitive **EPFLInnovators fellowship** funded from the European Union's Horizon 2020 research and innovation program for the doctoral degree

2015 – 2018: Ranking 3rd, MSc in computer science, Sharif Univ. of Tech.

2011 – 2016: Research fellowship from "National Elites Foundation"

2011 – 2015: Ranking in top 5%, BSc in electrical engineering, Sharif Univ. of Tech.

2011: Ranking 49th in the nation-wide university Entrance Exam

Computer skills

ML: PyTorch, Tensorflow, MATLAB, Knime, Scikit-learn

Programming: Python, CUDA, C, C++, MPI

Robotics: 8051 Assembly, Code Vision, Keil, Altium Designer, Proteus

Position of Responsibility

2023: Corresponding organizer of the JRDB workshop, in conjunction with ICCV, *Web-page*
designed a new challenge and benchmark for end-to-end motion forecasting in crowds

2021-2023: Peer review service: CVPR, ICCV, ICRA, ECCV, TR_C and TPAMI

2019-2022: Student committee member of the doctoral program, EPFL

2020: Co-organizing a weekly reading group on computer vision topics, EPFL, *Web-page*

Research Mentoring

Spring 2023: T. Trinca, "Accurate human motion forecasting: a certified approach"

Fall 2022: F. Forghani, "Realistic human motion prediction"

Spring 2022: Y. Gao, "Trajectory forecasting using visual input cues"

Fall 2021: C. Li, "3D human bounding box prediction in the wild"

Spring 2021: Y. Luo, "Facades segmentation and envelope type detection"

Fall 2020: M. Ghorbani, "Context-aware human image synthesis in the wild"

Teaching Assistantship

2019-2023: Deep Learning for Autonomous Vehicles, EPFL
designed and instructed exercise sessions, course project and was the guest lecturer on **generative models**

2019: Topics in Autonomous Robotics, EPFL

2018: Multi-GPU Tensorflow workshop, Sharif Univ. of Tech.
designed and instructed

2018: Introduction to Machine Learning, Sharif Univ. of Tech.

2017: Numerical Optimization, Sharif Univ. of Tech.

Languages

C2: English: Full professional proficiency

A2: French: Elementary proficiency

A1: German: Elementary proficiency

Extracurricular Activities

Playing and watching football

Hiking and exploring nature