RecTTA: RECONSTRUCTION-BASED TEST-TIME ADAPTATION FOR ROBUST TRAJECTORY PREDICTION IN DYNAMIC ENVIRONMENTS

Chiranthana Ramalingappa Rampura

A Thesis
in
The Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Master's (Computer Science) at
Concordia University
Montréal, Québec, Canada

Sep 2025

Concordia University

School of Graduate Studies

This is to certify that the thesis prepared

By: **Chiranthana Ramalingappa Rampura**

Entitled: **RecTTA: RECONSTRUCTION-BASED TEST-TIME ADAPTATION FOR ROBUST TRAJECTORY PREDICTION IN DYNAMIC ENVIRONMENTS**

and submitted in partial fulfillment of the requirements for the degree of

Master's (Computer Science)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

_____ Chair
Dr. Yiming Xiao

_____ Examiner
Dr. Yiming Xiao

_____ Examiner
Dr. Mirco Ravanelli

_____ Supervisor
Dr. Yang Wang

_____ Co-supervisor
Dr. Xinxin Zuo

Approved by _____
Dr. Graduate Program Director Name, Graduate Program Director

Sep 2025 _____
Dr. Dean Name, Dean
Faculty of Engineering and Computer Science

Abstract

RecTTA: RECONSTRUCTION-BASED TEST-TIME ADAPTATION FOR ROBUST TRAJECTORY PREDICTION IN DYNAMIC ENVIRONMENTS

Chiranthana Ramalingappa Rampura

The fundamental brittleness of trajectory prediction systems poses a critical challenge to the deployment of autonomous vehicles and robotic systems in dynamic real-world environments. While transformer-based models demonstrate exceptional performance on benchmark datasets, their static inference paradigm renders them vulnerable to distribution shifts—changes in environmental conditions, sensor characteristics, and motion patterns that inevitably arise during deployment. This limitation undermines reliability precisely when accurate predictions are most critical for safety-critical autonomous systems.

This thesis introduces **RecTTA (Reconstruction-based Test-Time Adaptation)**, a groundbreaking framework that fundamentally transforms trajectory prediction from static inference to dynamic adaptation. Unlike existing domain-specific adaptation methods, RecTTA leverages input trajectory reconstruction as a universal self-supervised signal that naturally preserves the spatial-temporal dependencies essential for motion forecasting. Through joint training with an auxiliary reconstruction decoder, our approach enables transformer models to continuously refine their internal representations for each test sample without requiring ground truth supervision.

Our comprehensive evaluation on the JTA dataset reveals three paradigm-shifting discoveries. First, RecTTA achieves consistent and substantial performance improvements of 4.09% ADE and 3.07% FDE reduction across diverse scenarios, establishing reconstruction-based adaptation as a robust enhancement mechanism. Second, we make a **counter-intuitive discovery** that fundamentally challenges conventional adaptation wisdom: selective adaptation of only the final output layers dramatically outperforms full model adaptation (3.55% vs 3.35% ADE improvement) while requiring 64.0% less computation time. This finding reveals that adaptation effectiveness concentrates in the prediction bottleneck rather than deep feature representations. Third, we demonstrate a **democratizing effect** where trajectory-only inputs achieve the largest relative improvements (7.50% ADE, 9.57% FDE), enabling resource-constrained systems to approach the performance of complex multi-modal configurations.

These contributions establish test-time adaptation as an essential paradigm for robust autonomous system deployment. By proving that strategic adaptation can simultaneously enhance performance

and computational efficiency while democratizing access to advanced prediction capabilities, this work provides both theoretical insights and practical tools for the next generation of adaptive AI systems. The principles underlying RecTTA—reconstruction as universal adaptation signal, selective parameter optimization, and architecture-aware adaptation strategies—extend beyond trajectory prediction to establish foundational concepts for adaptive neural architectures in structured prediction tasks.

Statement of Originality

I hereby declare that I am the sole author of this thesis. All ideas and inventions attributed to others have been properly referenced. I understand that my thesis may be made electronically available to the public.

Acknowledgments

I would like to express my deepest gratitude to all those who have supported and guided me throughout this research journey.

First and foremost, I extend my sincere appreciation to my supervisor, Dr. Yang Wang, for his invaluable guidance, continuous support, and unwavering patience throughout this research. His expertise in this field has been instrumental in shaping this work. Dr. Wang's insightful feedback, constructive criticism, and encouragement have not only enhanced the quality of this thesis but have also contributed significantly to my growth as a researcher.

I am equally grateful to my co-supervisor, Dr. Xinxin Zuo, for her exceptional mentorship and technical insights. Her expertise in computer vision and her meticulous attention to detail have been crucial in refining the methodological aspects of this research. Dr. Zuo's collaborative approach and thoughtful discussions have enriched my understanding of the field and improved the rigor of this work.

Finally, I am profoundly grateful to my parents, both esteemed physics professors, whose lifelong dedication to scientific inquiry has been my greatest inspiration. Their constant support, encouragement, and gentle yet persistent motivation have been the foundation of my academic pursuits. Their belief in my abilities, even during the most challenging moments of this journey, has been my source of strength and determination.

This thesis would not have been possible without the collective support, guidance, and inspiration from all these remarkable individuals.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Background

Human trajectory prediction stands as a fundamental challenge at the intersection of computer vision, robotics, and artificial intelligence. This task—forecasting the future positions of individuals based on their past movements and contextual information—has far-reaching implications across numerous domains. In autonomous driving systems, accurate trajectory forecasting enables vehicles to anticipate pedestrian movements and execute safe navigation decisions [30]. For surveillance applications, it facilitates early detection of unusual behaviors and crowd management [1]. In human-robot interaction scenarios, it allows robots to navigate social spaces while respecting personal boundaries and social norms [20]. The breadth of these applications underscores the critical importance of developing robust trajectory prediction models.

The challenge of trajectory prediction is compounded by the inherently stochastic nature of human motion. Unlike rigid objects that follow deterministic physics, human movement is influenced by a complex interplay of factors: physical constraints of the environment, social dynamics between individuals, personal goals and intentions, and cultural norms governing movement in shared spaces [31]. As Kothari et al. [19] observe, "human trajectory forecasting requires understanding both explicit physical constraints and implicit social conventions—a nuanced interplay that remains challenging for computational models to capture fully."

This complexity manifests in several ways. First, human motion is fundamentally multimodal—given the same historical trajectory, multiple future paths may be equally valid depending on the pedestrian's intentions. Second, motion is contextual, with decisions influenced by environmental features, nearby agents, and dynamic obstacles. Third, trajectories exhibit both short-term physical constraints (momentum, maximum acceleration) and long-term goal-directed behavior, requiring

models to reason across multiple time scales simultaneously [24].

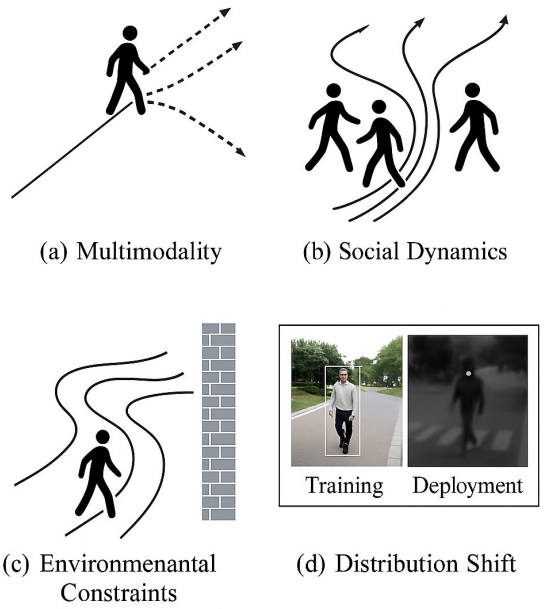


Figure 1: Illustration of key challenges in human trajectory prediction: (a) **Multimodality**: multiple plausible future paths from the same history, (b) **Social dynamics**: interactions between pedestrians affecting movement patterns, (c) **Environmental constraints**: physical obstacles and designated paths, and (d) **Distributional shift**: differences between training and deployment conditions.

Trajectory Prediction: Formal Definition. Formally, the trajectory prediction task can be defined as follows: Given an observed sequence of positions $\mathbf{X}_{1:t} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_t\}$ for a pedestrian up to time t, the objective is to predict their future positions $\mathbf{X}_{t+1:t+\tau} = \{\mathbf{x}_{t+1}, \mathbf{x}_{t+2}, ..., \mathbf{x}_{t+\tau}\}$ for the next τ timesteps. Each position $\mathbf{x}_i$ typically represents a 2D location (x, y) in the ground plane at time i. In more sophisticated frameworks, this representation is augmented with additional cues such as 3D poses (capturing body articulation), bounding boxes (providing scale and orientation), and social context (positions and movements of nearby agents).

The predominant approach to this problem has evolved from hand-crafted models based on social forces [14] to data-driven deep learning techniques. Early deep learning models employed recurrent architectures like LSTMs to capture temporal dependencies [1, 12], while more recent approaches have leveraged the power of attention mechanisms and transformers to model complex spatio-temporal relationships [45, 37]. These advances have progressively improved prediction accuracy on benchmark datasets, with state-of-the-art models achieving impressive performance under controlled conditions.

However, a critical limitation persists across these approaches: they operate under what can be characterized as a "closed-world assumption." This assumption posits that the distribution of data encountered during deployment will closely match that of the training data—an assumption that rarely holds in practice. Real-world deployment environments often present significant distributional shifts from training conditions, including variations in lighting, weather, camera viewpoints, crowd density, and cultural contexts governing movement [15]. These shifts can severely degrade model performance, undermining the reliability of trajectory predictions in precisely the high-stakes scenarios where accuracy is most crucial.

1.2 Challenges in Real-World Trajectory Prediction

1.2.1 The Problem of Static Inference

Deep neural networks, despite their remarkable representational capacity, suffer from a fundamental limitation: once trained, their parameters remain static during inference. This characteristic, while computationally efficient, renders these models inherently brittle when confronted with inputs that deviate from their training distribution. As Sun et al. [36] note, "Neural networks encode a fixed understanding of the world based on their training data, leaving them vulnerable to even subtle distribution shifts."

This vulnerability is particularly pronounced in trajectory prediction systems deployed in dynamic real-world environments. Consider an autonomous vehicle trained on data collected primarily in clear weather conditions—when deployed in fog or heavy rain, the quality and characteristics of sensor inputs may change dramatically. Similarly, a model trained on data from one cultural context may fail to generalize to regions with different pedestrian behaviors and social norms. Even within the same deployment environment, temporary occlusions, sensor malfunctions, or unusual lighting conditions can produce inputs that fall outside the model's training distribution.

Distribution Shift: Taxonomy and Impact. The concept of distribution shift encompasses several distinct phenomena that affect model performance in different ways:

- *Covariate Shift*: This occurs when the distribution of input features changes between training and deployment, while the conditional relationship between inputs and outputs remains constant. In trajectory prediction, covariate shift might manifest as changes in camera angle, sensor noise characteristics, or environmental conditions that alter the appearance or quality of input features without changing the fundamental physics of human motion.

- *Label Shift*: This refers to changes in the marginal distribution of output variables. In the context of trajectory prediction, this might involve deployment in environments with different

frequencies of motion patterns or destinations—for instance, a model trained in a shopping mall being deployed in an airport terminal, where the distribution of walking speeds and destinations differs significantly.

- *Concept Shift*: Perhaps the most challenging form of distribution shift, concept shift involves changes in the relationship between inputs and outputs. This might occur when social norms governing movement change across cultures or contexts, altering how pedestrians respond to the same environmental cues.

- *Temporal Shift*: A special case relevant to trajectory prediction, temporal shift involves gradual changes in data distribution over time, such as seasonal variations in pedestrian behavior or evolving social norms.

These various forms of distribution shift can compound each other in real-world deployments, creating complex challenges for static models. Without mechanisms for adaptation, such models continue to make predictions based on assumptions that may no longer hold, potentially leading to cascading errors in downstream decision-making systems. This is particularly concerning in safety-critical applications like autonomous driving, where erroneous trajectory predictions could lead to collision risks.

Empirical studies have demonstrated the severity of this problem. Hendrycks and Dietterich [15] showed that even state-of-the-art deep learning models can experience dramatic performance degradation when tested on corrupted versions of their training data. In the specific context of trajectory prediction, Mangalam et al. [23] observed performance drops of up to 40% when models trained on one dataset were evaluated on another, highlighting the brittleness of these systems to distribution shifts.

1.2.2 Multi-Modal Cue Dependency and Vulnerability

Recent advances in trajectory prediction have increasingly leveraged multi-modal approaches that incorporate rich contextual cues beyond simple positional data. State-of-the-art models like Social-Transmotion [37] integrate multiple input modalities—trajectory history, 3D human poses, 2D poses, and bounding boxes—through sophisticated transformer architectures. This multi-modal approach has yielded significant performance improvements under ideal conditions, with the model learning to fuse complementary information across modalities.

However, this reliance on multiple input cues introduces a critical vulnerability: if any modality is corrupted, missing, or significantly different from training examples, the model's performance can degrade substantially. This vulnerability is particularly acute in real-world deployments, where various factors can compromise input quality:

- *Occlusions*: Partial or complete occlusions can render certain modalities temporarily unavailable. For instance, a pedestrian walking behind an obstacle might have their lower body occluded, making leg keypoints unavailable for pose estimation.
- *Sensor Limitations*: Different sensors have varying capabilities and failure modes. Depth sensors may struggle in bright sunlight, while RGB cameras might provide degraded inputs in low-light conditions.
- *Processing Artifacts*: The preprocessing pipeline for extracting features like poses and bounding boxes can introduce its own errors and artifacts, especially when underlying computer vision models encounter edge cases.
- *Novel Scenarios*: Deployment environments may present novel combinations of factors not seen during training, such as unusual clothing, carrying objects, or movement patterns specific to particular cultural contexts.

Cue Masking: A Partial Solution. To address the challenge of modality corruption or absence, Social-Transmotion and similar approaches employ a technique called cue masking during training. This involves randomly dropping out entire modalities or portions of each modality with certain probabilities:

- $p_{modality}$: The probability of masking an entire modality (typically 0.3)
- p_{traj}: The probability of masking trajectory data (typically 0.1)
- p_{joint}: The probability of masking individual joints in pose data (typically 0.1)

This approach forces the model to learn robust representations that can function even when some inputs are missing, essentially training the model to rely on whatever cues are available. While this strategy improves robustness to missing modalities, it has a fundamental limitation: it still results in a static model at inference time. The model cannot dynamically recalibrate its reliance on different cues based on their quality or reliability in a specific test instance.

Consider a scenario where a pedestrian's 3D pose estimation is temporarily corrupted due to unusual lighting conditions. A model trained with cue masking might have learned a general strategy for handling missing pose data, but it cannot specifically adapt its internal representations to the particular pattern of corruption present in this specific instance. The model's attention weights and feature extraction remain fixed, potentially leading to suboptimal predictions when the corruption pattern differs from those encountered during training.

This limitation highlights a broader challenge in deep learning: the trade-off between generalization and specialization. Models trained to handle a wide variety of corruption patterns (through

techniques like cue masking) may not be optimal for any specific pattern. This suggests the need for approaches that can dynamically adapt to each test instance's unique characteristics—precisely the motivation behind test-time adaptation techniques.

1.3 Test-Time Adaptation: Principles and Potential

Test-Time Adaptation (TTA) represents a paradigm shift in how we approach the deployment of deep learning models. Rather than treating a trained model as a static entity, TTA conceptualizes it as a dynamic system capable of adapting to each test input individually. This adaptation occurs during inference, without requiring ground truth labels or extensive retraining on new data.

Formal Definition of TTA. Test-Time Adaptation can be formally defined as follows: Given a model f_θ with parameters θ trained on a source distribution $P_{source}(X, Y)$, and a test sample x_{test} from a potentially different distribution $P_{target}(X, Y)$, TTA involves updating a subset of the model parameters $\theta' \subseteq \theta$ using only x_{test} to improve performance on this specific sample. This adaptation is typically guided by a self-supervised objective $\mathcal{L}_{self}$ that does not require access to the ground truth label y_{test}.

The key insight enabling TTA is that even without ground truth labels, certain properties of the input data can provide useful learning signals. These self-supervised signals allow the model to align its internal representations with the current test input, potentially correcting for distribution shifts on a per-sample basis.

Prior Work in TTA. Several approaches to TTA have emerged in recent literature, primarily focused on image classification tasks:

- *TENT* [40] adapts batch normalization statistics and parameters by minimizing the entropy of the model's predictions. This approach is computationally efficient but limited to updating only a small subset of model parameters.

- *TTT (Test-Time Training)* [36] employs a self-supervised rotation prediction task to adapt encoder representations. During training, the model learns both the primary task and the rotation prediction task using a shared encoder. At test time, only the rotation prediction loss is used to update encoder parameters.

- *SHOT* [21] focuses on unsupervised domain adaptation by using pseudo-labeling and information maximization to adapt classifier-level representations without requiring source data access.

- *MEMO* [46] leverages test-time augmentations and consistency regularization to adapt model

parameters, encouraging consistent predictions across different augmented versions of the same input.

While these approaches have shown promising results in image classification domains, they face significant challenges when applied to sequence data like trajectories. Temporal dependencies, multi-modal inputs, and the predictive nature of trajectory forecasting create unique requirements that are not addressed by existing TTA frameworks.

For instance, entropy minimization (as used in TENT) may not be appropriate for trajectory prediction, where the goal is to capture the inherent multimodality of human motion rather than produce confident single-path predictions. Similarly, rotation prediction (as used in TTT) is not directly applicable to trajectory data, which lacks the spatial structure of images. These limitations highlight the need for TTA approaches specifically designed for trajectory prediction tasks.

1.4 RecTTA: Reconstruction-based Test-Time Adaptation

To address the unique challenges of test-time adaptation for trajectory prediction, we introduce **RecTTA (Reconstruction-based Test-Time Adaptation)**, a novel approach that leverages input reconstruction as a self-supervised signal for adaptation. RecTTA extends transformer-based trajectory prediction models by adding an auxiliary decoder branch trained to reconstruct the input sequences from the shared encoder representations.

Key Insights and Design Principles. The design of RecTTA is guided by several key insights:

1. *Reconstruction as a Universal Self-Supervised Signal*: Unlike task-specific self-supervised objectives, reconstruction applies naturally to any input modality and preserves both spatial and temporal structure. By training a model to reconstruct its inputs, we encourage it to retain detailed information about the input distribution in its latent representations.

2. *Shared Encoder, Dual Decoders*: By using a shared encoder for both reconstruction and prediction tasks, we ensure that improvements in representation quality benefit both objectives. The primary decoder focuses on future prediction, while the auxiliary decoder specializes in reconstruction.

3. *Selective Parameter Updates*: During test-time adaptation, we update only the encoder parameters while keeping the prediction decoder frozen. This selective update strategy preserves the model's predictive capabilities while allowing its representations to align with the current test input.

4. *Per-Sample Adaptation*: Rather than adapting to an entire target domain, RecTTA adapts to each test sample individually. This fine-grained adaptation allows the model to handle diverse and unpredictable distribution shifts encountered in real-world deployments.

1.5 Research Questions and Contributions

This thesis addresses three fundamental research questions that challenge conventional approaches to trajectory prediction and test-time adaptation:

1. **Can reconstruction-based self-supervision enable effective test-time adaptation for trajectory prediction without ground truth labels?** This question explores whether input reconstruction can serve as a universal adaptation signal that preserves the temporal and spatial structure essential for motion prediction, enabling robust deployment in unseen environments.

2. **Does selective parameter adaptation outperform full model adaptation in transformer-based trajectory prediction?** This investigates the counter-intuitive hypothesis that strategic constraint of adaptation might simultaneously improve both performance and computational efficiency, challenging conventional wisdom about parameter updating in test-time adaptation.

3. **Can test-time adaptation democratize trajectory prediction by enabling simple input configurations to approach multi-modal performance?** This examines whether adaptive mechanisms can bridge the performance gap between resource-constrained and fully-equipped systems, making advanced trajectory prediction accessible across diverse deployment scenarios.

In addressing these questions, this thesis makes the following key contributions:

- **RecTTA Framework**: We introduce **RecTTA (Reconstruction-based Test-Time Adaptation)**, the first framework specifically designed for trajectory prediction that leverages input reconstruction as a universal self-supervised adaptation signal. This achieves substantial performance improvements of 4.09% ADE and 3.07% FDE reduction over state-of-the-art baselines while requiring no ground truth labels during adaptation.

- **Selective Adaptation Discovery**: We make a **counter-intuitive discovery** that selective adaptation of only the final output layers achieves superior performance (3.55% ADE improvement) while reducing computational overhead by 64.1% compared to full model adaptation.

This finding fundamentally challenges conventional wisdom about parameter updating in test-time adaptation, revealing that deep transformer layers may not be the primary drivers of adaptation effectiveness. Instead, our results demonstrate that adaptation primarily occurs through fine-tuning of the final prediction layers, achieving the best performance-to-efficiency trade-off with adaptation times of only 0.245 seconds per batch compared to 0.681 seconds for full model adaptation.

- **Democratization Effect**: We demonstrate a **democratizing effect** where trajectory-only inputs benefit most dramatically from adaptation (7.50% ADE, 9.57% FDE improvements), enabling resource-constrained systems to approach the performance of complex multi-modal setups. This makes advanced trajectory prediction accessible across diverse deployment scenarios, from simple sensors to fully-equipped autonomous systems.

- **Comprehensive Evaluation Framework**: We develop a **systematic evaluation methodology** encompassing modality analysis, error regime studies, and motion pattern evaluation that provides nuanced understanding of when and why test-time adaptation succeeds or fails in trajectory prediction tasks. This framework establishes adaptation dynamics principles through systematic ablation studies, identifying optimal convergence at 3 adaptation steps and demonstrating the effectiveness of moderate learning rates (0.0005) for stable adaptation behavior.

These contributions collectively establish test-time adaptation as an essential strategy for robust trajectory prediction deployment. By demonstrating that strategic adaptation can simultaneously improve performance and computational efficiency while making advanced prediction capabilities accessible to resource-constrained systems, this work fundamentally advances our understanding of adaptive neural architectures and provides practical tools for real-world autonomous system deployment.

1.6 Thesis Structure

This thesis presents a systematic investigation of reconstruction-based test-time adaptation for human trajectory prediction, organized into the following chapters:

Chapter 2: Background and Related Work establishes the theoretical foundation by reviewing the evolution of trajectory prediction methods from classical social force models to modern transformer architectures. It critically examines the limitations of static inference in real-world deployments, surveys existing test-time adaptation approaches, and identifies the specific challenges in applying these techniques to multi-modal trajectory prediction. This chapter contextualizes RecTTA

within the broader research landscape and highlights the unaddressed gap between sophisticated prediction models and their ability to adapt during deployment.

Chapter 3: Methodology details the technical innovations of our RecTTA framework. It presents the architectural extension of the Social-Transmotion model with auxiliary reconstruction capabilities, explains the joint training strategy that balances prediction and reconstruction objectives, and introduces the test-time adaptation mechanism that updates encoder parameters while keeping the primary prediction pathway frozen. This chapter provides the algorithmic foundations, formal optimization objectives, and theoretical justification for our approach to self-supervised adaptation in trajectory prediction tasks.

Chapter 4: Experimental Methodology and Results presents our comprehensive evaluation approach and the empirical validation of RecTTA. It begins by describing the JTA dataset configuration, evaluation metrics, and systematic experimental design encompassing adaptation dynamics, modality analysis, and architecture-aware studies. The chapter then presents detailed quantitative results revealing our counter-intuitive discovery that selective adaptation of output layers outperforms full model adaptation while significantly reducing computational overhead. It demonstrates the democratizing effect of our approach on resource-constrained systems and provides comprehensive ablation studies that isolate the contribution of each component to overall performance. The chapter concludes with qualitative analyses of motion patterns and error distributions, establishing the boundaries of effective test-time adaptation in trajectory prediction.

Chapter 5: Discussion and Conclusion synthesizes our empirical findings into fundamental principles about adaptation in structured prediction tasks and provides a comprehensive conclusion to the thesis. It examines the theoretical implications of our core contributions, compares RecTTA with state-of-the-art methods, addresses limitations related to pattern sensitivity and over-adaptation, and outlines promising future research directions including meta-learning integration, uncertainty-aware adaptation, and multi-modal frameworks. The chapter concludes by establishing the broader impact of our work and emphasizing how RecTTA transforms trajectory prediction from static to adaptive systems, enabling robust performance across distribution shifts without requiring ground truth labels.

Chapter 2

Background and Related Work

2.1 Trajectory Prediction in Dynamic Environments

Human trajectory prediction has evolved significantly over the past decade, transitioning from hand-crafted models based on social forces [14] to sophisticated deep learning architectures. This evolution reflects the increasing recognition of the complex, multimodal nature of human motion and the need to incorporate rich contextual information for accurate predictions.

2.1.1 Traditional Approaches

Early deep learning approaches to trajectory prediction primarily relied on recurrent neural networks (RNNs) to capture the temporal dependencies in human motion. Social-LSTM [1] pioneered the integration of social context by introducing a "social pooling" layer that allowed information sharing between nearby pedestrians' LSTMs. This approach was extended by Social-GAN [12], which employed a generative adversarial framework to capture the multimodal nature of future trajectories and produce diverse, socially acceptable predictions.

Subsequent work explored various mechanisms for modeling social interactions. Social-STGCNN [25] employed spatio-temporal graph convolutional networks to model pedestrian interactions as a graph, where nodes represent pedestrians and edges capture their spatial relationships. Trajectron++ [33] introduced a modular, graph-structured recurrent model that incorporated both agent dynamics and heterogeneous scene context.

These approaches demonstrated increasing sophistication in modeling social interactions and environmental constraints. However, they typically relied on relatively simple input representations—primarily 2D positions—and struggled to fully leverage the rich multimodal cues available in video data, such as human poses and detailed scene context.

2.1.2 Transformer-based Methods

The introduction of transformer architectures [38] to trajectory prediction marked a significant advancement in the field. Transformers' ability to capture long-range dependencies through self-attention mechanisms made them particularly well-suited for modeling complex spatio-temporal relationships in human motion.

AgentFormer [45] was among the first to apply transformers to trajectory prediction, using a joint agent-time attention mechanism to model interactions across both the spatial and temporal dimensions simultaneously. This approach enabled more flexible modeling of variable-length interactions compared to recurrent architectures.

More recently, Social-Transmotion [37] introduced a dual-transformer architecture specifically designed for multi-modal trajectory prediction. This architecture consists of:

- A Cross-Modality Transformer (CMT) that integrates multiple input cues (trajectory, 3D/2D poses, 3D/2D bounding boxes) for each agent individually
- A Social Transformer (ST) that models interactions between agents by attending to their encoded representations

Social-Transmotion demonstrated state-of-the-art performance on several benchmark datasets, highlighting the benefits of integrating multiple input modalities through transformer architectures. The model's ability to handle variable numbers of agents and flexible input representations made it particularly well-suited for complex, real-world scenarios.

Other notable transformer-based approaches include STAR [44], which employed a spatio-temporal transformer to jointly model spatial interactions and temporal dependencies, and MemoNet [43], which incorporated memory mechanisms to capture long-term motion patterns.

2.1.3 Persistent Challenges

Despite these advances, several challenges persist in trajectory prediction:

- **Multimodality**: Human motion is inherently multimodal—given the same history, multiple future paths may be equally valid. While generative approaches like Social-GAN [12] and variational methods like Trajectron++ [33] attempt to capture this multimodality, accurately representing the distribution of possible futures remains challenging.
- **Occlusions and Missing Cues**: In real-world scenarios, input cues may be partially or completely occluded. While approaches like Social-Transmotion employ cue masking during training to improve robustness, they cannot dynamically adapt to specific patterns of occlusion encountered during deployment.

- **Long-term Prediction**: Accurately predicting trajectories over extended time horizons becomes increasingly difficult due to the accumulation of uncertainty and the influence of unobserved factors like changing goals or intentions.

- **Distribution Shifts**: Perhaps most critically, models trained on specific datasets may fail to generalize to new environments or scenarios with different characteristics. This challenge, which we discuss in detail in the next section, motivates our work on test-time adaptation.

As noted by Rudenko et al. [31] in their comprehensive survey, "Despite significant progress, trajectory prediction in complex, dynamic environments with heterogeneous agents remains an open challenge, particularly when deployed in previously unseen contexts." This observation highlights the need for approaches that can adapt to new environments and scenarios without requiring extensive retraining.

2.2 Distribution Shift and the Closed-World Assumption

Deep learning models, including state-of-the-art trajectory prediction systems, typically operate under what is known as the "closed-world assumption"—the expectation that test data will follow the same distribution as training data. This assumption is formalized in the standard supervised learning framework, where both training and test samples are assumed to be drawn independently from the same underlying distribution $P(X, Y)$.

However, as Hendrycks and Dietterich [15] demonstrate, "this assumption rarely holds in practice, as real-world data often deviates significantly from the training distribution." This deviation, known as distribution shift, poses a fundamental challenge to the deployment of machine learning systems in real-world environments.

2.2.1 Types of Distribution Shift

Distribution shift can manifest in various forms, each presenting unique challenges for trajectory prediction models:

- **Covariate Shift**: This occurs when the distribution of input features changes between training and deployment, while the conditional relationship between inputs and outputs remains constant: $P_{train}(X) \neq P_{test}(X)$ but $P_{train}(Y|X) = P_{test}(Y|X)$. In trajectory prediction, covariate shift might manifest as changes in camera viewpoint, lighting conditions, or sensor characteristics that alter the appearance or quality of input features.

- **Label Shift**: This involves changes in the marginal distribution of output variables: $P_{train}(Y) \neq P_{test}(Y)$ but $P_{train}(X|Y) = P_{test}(X|Y)$. In trajectory prediction, this might involve deployment in environments with different frequencies of movement patterns or destinations—for instance, a model trained in a shopping mall being deployed in an airport terminal.

- **Concept Shift**: This most challenging form of distribution shift involves changes in the relationship between inputs and outputs: $P_{train}(Y|X) \neq P_{test}(Y|X)$. This might occur when social norms governing movement change across cultures or contexts, altering how pedestrians respond to the same environmental cues.

- **Temporal Shift**: Particularly relevant to trajectory prediction, temporal shift involves gradual changes in data distribution over time, such as seasonal variations in pedestrian behavior or evolving social norms.

2.2.2 Impact on Trajectory Prediction

The impact of distribution shift on trajectory prediction models can be severe. Mangalam et al. [23] observed performance drops of up to 40% when models trained on one dataset were evaluated on another, highlighting the brittleness of these systems to distribution shifts. Similarly, Rasouli et al. [30] documented significant degradation in pedestrian behavior prediction when models were deployed in novel environmental contexts.

These challenges are particularly acute for multi-modal approaches like Social-Transmotion [37] that rely on multiple input cues. If any modality experiences distribution shift—for instance, if pose estimation quality degrades due to different lighting conditions—the model's performance can suffer dramatically. While techniques like cue masking during training provide some robustness to missing modalities, they cannot address the specific patterns of corruption or shift encountered during deployment.

As Quionero-Candela et al. [29] note in their seminal work on dataset shift, "The reality of machine learning is that training and test data are often drawn from different distributions, and this can dramatically impact the performance of deployed systems." This observation underscores the need for approaches that can adapt to distribution shifts during deployment—precisely the motivation behind test-time adaptation techniques.

2.3 Test-Time Adaptation (TTA)

Test-Time Adaptation (TTA) represents a paradigm shift in how we approach the deployment of deep learning models. Rather than treating a trained model as a static entity, TTA conceptualizes

it as a dynamic system capable of adapting to each test input individually. This adaptation occurs during inference, without requiring ground truth labels or extensive retraining on new data.

2.3.1 General TTA Techniques in Computer Vision

The field of test-time adaptation has seen significant development in recent years, particularly in computer vision applications. Several key approaches have emerged:

- **Entropy Minimization**: TENT [40] adapts batch normalization statistics and parameters by minimizing the entropy of the model's predictions. The underlying intuition is that confident predictions (low entropy) are more likely to be correct, even under distribution shift. By updating model parameters to minimize prediction entropy, TENT encourages the model to make more confident predictions on test data.

- **Information Maximization**: SHOT [21] employs information maximization and pseudo-labeling to adapt classifier-level representations without requiring source data access. The approach encourages feature alignment between source and target domains while maintaining discriminative power.

- **Self-Supervised Tasks**: Test-Time Training (TTT) [36] leverages a self-supervised rotation prediction task to adapt encoder representations. During training, the model learns both the primary task and the rotation prediction task using a shared encoder. At test time, only the rotation prediction loss is used to update encoder parameters.

- **Consistency Regularization**: MEMO [46] uses test-time augmentations and consistency regularization to adapt model parameters, encouraging consistent predictions across different augmented versions of the same input.

These approaches have demonstrated impressive results in image classification tasks, significantly improving model robustness to various forms of distribution shift. However, they face challenges when applied to sequence data like trajectories, where temporal dependencies and the predictive nature of the task create unique requirements.

2.3.2 TTA for Video and Sequence Data

Extending test-time adaptation to video and sequence data introduces additional challenges related to temporal consistency and the dynamic nature of the input. Several approaches have been developed to address these challenges:

- **TempT** [35] enforces temporal consistency in video adaptation by incorporating a temporal consistency loss that encourages similar predictions for temporally adjacent frames. This approach leverages the natural smoothness of video data to guide adaptation.
- **Video-TTA** [4] employs contrastive learning across frames to adapt feature representations, treating temporally adjacent frames as positive examples and distant frames as negative examples. This approach encourages the model to learn temporally consistent features.
- **T4P** [28] specifically addresses trajectory prediction by using a masked autoencoder approach combined with actor-specific token memory to adapt deep layers during inference. The model employs a regression loss on predicted trajectories to guide adaptation, focusing on improving prediction accuracy directly.

While these approaches represent important steps toward adapting sequence models at test time, they often rely on task-specific objectives or require complex architectural modifications. A more general, lightweight approach to test-time adaptation for trajectory prediction remains an open challenge.

2.3.3 Theoretical Underpinnings of TTA

The effectiveness of test-time adaptation can be understood through several theoretical lenses:

- **Domain Adaptation Theory**: From the perspective of domain adaptation, TTA can be viewed as an online, instance-specific adaptation process that minimizes the discrepancy between source and target domains [2]. By updating model parameters to better fit each test instance, TTA reduces the domain gap on a per-sample basis.
- **Self-Supervised Learning**: TTA leverages principles from self-supervised learning, where models learn useful representations from the data itself without requiring explicit labels [5]. By employing self-supervised objectives like reconstruction or consistency, TTA enables adaptation without ground truth labels.
- **Continual Learning**: TTA can also be connected to continual learning, where models must adapt to new data while preserving performance on previously learned tasks [18]. The challenge of catastrophic forgetting—where adaptation to new data degrades performance on the original task—is particularly relevant for TTA, motivating approaches that selectively update parameters or employ regularization techniques.

These theoretical perspectives provide valuable insights into the design of effective TTA approaches, guiding decisions about which parameters to update, what objectives to optimize, and how to balance adaptation with stability.

Table 1: Comparison of Test-Time Adaptation Methods

Method	**Domain**	**Labels Required**	**Adaptation Signal**	**Target Module**	**Comments**
TENT [40]	Image Classification	No	Entropy Minimization	Batch Norm Stats	Shallow adaptation only
SHOT [21]	Image Classification	No	Information Maximization	Classifier	Not suitable for sequences
TTT [36]	Image Classification	No	Rotation Prediction	Encoder	Requires auxiliary task
MEMO [46]	Image Classification	No	Consistency Regularization	Full Model	Augmentation-based
TempT [35]	Video	No	Temporal Consistency	Encoder	Video-specific constraints
Video-TTA [4]	Video	No	Contrastive Learning	Encoder	Frame alignment focus
T4P [28]	Trajectory	No	MAE + Regression	Decoder & Encoder	Complex architecture
Point-TTA [13]	Point Cloud	No	Reconstruction	Encoder	Multi-task adaptation
RecTTA (Ours)	Trajectory	No	**Reconstruction**	**Encoder**	**Lightweight, sequence-aware**

2.4 Auxiliary Tasks and Representation Learning

Auxiliary tasks have emerged as a powerful tool for improving representation learning in deep neural networks. By training models to perform multiple related tasks simultaneously, auxiliary learning encourages the development of more robust and generalizable representations that capture underlying structure in the data.

2.4.1 Types of Auxiliary Tasks

Various types of auxiliary tasks have been explored in the literature, each offering different benefits for representation learning:

- **Reconstruction**: Training models to reconstruct their inputs from latent representations encourages the preservation of detailed information about the input distribution. Autoencoders [16] and their variants exemplify this approach, with applications ranging from image compression to representation learning.

- **Denoising**: Denoising tasks involve reconstructing clean inputs from corrupted versions, encouraging models to learn robust features that capture underlying structure while ignoring noise [39]. This approach is particularly valuable for improving robustness to input corruption.

- **Rotation/Transformation Prediction**: Tasks like predicting image rotations [9] or other geometric transformations encourage models to learn semantic features that capture object structure and orientation.

- **Jigsaw Puzzles**: Training models to reassemble shuffled patches of inputs [27] encourages the learning of spatial relationships and contextual understanding.

- **Contrastive Learning**: Tasks that involve distinguishing between similar and dissimilar examples [5] encourage models to learn representations that capture semantic similarity while ignoring irrelevant variations.

2.4.2 Benefits for Trajectory Prediction

Auxiliary tasks offer several specific benefits for trajectory prediction models:

- **Improved Feature Learning**: By providing additional learning signals, auxiliary tasks can help models discover more informative features that capture underlying patterns in human motion. Xu et al. [41] demonstrated that incorporating reconstruction and denoising tasks significantly improved the quality of learned representations for spatio-temporal modeling.

- **Robustness to Missing Data**: Auxiliary tasks like reconstruction can improve model robustness to missing or corrupted inputs. By learning to reconstruct complete inputs from partial observations, models develop representations that are less sensitive to specific input features.

- **Regularization**: Auxiliary tasks can serve as a form of regularization, preventing overfitting to the primary task and encouraging more generalizable representations. This is particularly valuable in trajectory prediction, where training data may be limited or biased.

- **Self-Supervised Adaptation**: Perhaps most relevant to our work, auxiliary tasks provide self-supervised signals that can guide adaptation during inference. By training models to perform tasks that don't require ground truth labels, we enable adaptation in scenarios where such labels are unavailable.

As Caruana [3] noted in his seminal work on multitask learning, "MTL improves generalization by leveraging the domain-specific information contained in the training signals of related tasks."

This observation is particularly relevant for trajectory prediction, where the complex, multimodal nature of human motion benefits from rich, informative representations.

2.5 Integrating Auxiliary Learning with Test-Time Adaptation

The integration of auxiliary learning with test-time adaptation represents a promising direction for improving model robustness to distribution shifts. By training models to perform auxiliary tasks alongside their primary objective, we can establish self-supervised signals that enable adaptation during inference.

2.5.1 Meta-Auxiliary Learning

Meta-auxiliary learning [6] extends the concept of auxiliary tasks to the meta-learning framework, where models are explicitly trained to adapt using auxiliary task signals. This approach involves:

- Training a model to perform both primary and auxiliary tasks
- Simulating distribution shifts during training
- Teaching the model to adapt to these shifts using only the auxiliary task loss
- Evaluating adaptation performance on the primary task

This meta-optimization process essentially trains the model to "learn how to adapt" using auxiliary signals, preparing it for effective test-time adaptation in real-world scenarios.

Point-TTA [13] applies this concept to point cloud registration, using auxiliary tasks like reconstruction and feature alignment to adapt models at test time. The approach demonstrates significant improvements in registration accuracy under various distribution shifts, highlighting the potential of auxiliary-guided adaptation.

2.5.2 Reconstruction as an Adaptation Signal

Among various auxiliary tasks, reconstruction stands out as particularly well-suited for test-time adaptation in trajectory prediction:

- **Universality**: Reconstruction applies naturally to any input modality, making it suitable for multi-modal trajectory prediction models that process various types of inputs (positions, poses, bounding boxes).

- **Temporal Structure Preservation**: Reconstruction inherently preserves the temporal structure of sequence data, encouraging models to maintain temporal dependencies that are crucial for trajectory prediction.
- **Alignment with Input Distribution**: By optimizing for reconstruction quality, models naturally align their internal representations with the current input distribution, addressing covariate shift directly.
- **Computational Efficiency**: Reconstruction typically requires minimal additional parameters (often just a decoder mirroring the encoder structure), making it a lightweight addition to existing models.

These properties make reconstruction an ideal auxiliary task for guiding test-time adaptation in trajectory prediction models, as we demonstrate with our RecTTA approach.

2.6 Gaps in Existing Literature

Despite the significant advances in trajectory prediction and test-time adaptation, several important gaps remain in the existing literature:

- **Limited Exploration of TTA for Trajectory Prediction**: While test-time adaptation has been extensively studied for image classification and, to a lesser extent, video understanding, its application to trajectory prediction remains largely unexplored. The unique challenges of trajectory data—including temporal dependencies, multimodality, and the predictive nature of the task—require specialized approaches.
- **Lack of Lightweight Adaptation Methods**: Existing approaches to test-time adaptation often require complex architectural modifications or computationally expensive adaptation procedures. There is a need for lightweight, efficient adaptation methods that can be easily integrated into existing trajectory prediction models.
- **Insufficient Attention to Multi-Modal Inputs**: Many trajectory prediction models now incorporate multiple input modalities, but existing TTA approaches do not adequately address the challenges of adapting to distribution shifts across different modalities.
- **Static Attention in Transformer Models**: State-of-the-art transformer-based models like Social-Transmotion employ attention mechanisms to integrate information across modalities and agents, but these attention patterns remain fixed during inference. There is a need for approaches that can dynamically adapt attention based on input quality and reliability.

- **Limited Understanding of Adaptation Dynamics**: The dynamics of test-time adaptation for sequence models remain poorly understood, particularly regarding the impact of adaptation hyperparameters, the selection of parameters to update, and the interaction between adaptation and existing robustness techniques like cue masking.

Our work addresses these gaps by introducing RecTTA, a lightweight, reconstruction-based test-time adaptation approach specifically designed for transformer-based trajectory prediction models. By leveraging the power of auxiliary reconstruction to guide adaptation, RecTTA enables dynamic, per-sample adaptation without requiring ground truth labels or complex architectural modifications.

2.7 Relevance to Our Project

The literature review presented in this chapter highlights several key insights that inform our approach:

- **Transformer-Based Architectures**: State-of-the-art trajectory prediction models like Social-Transmotion employ transformer architectures to integrate multiple input modalities and model social interactions. Our work builds upon this foundation, extending Social-Transmotion with an auxiliary reconstruction branch.
- **Distribution Shift Challenges**: Real-world deployment of trajectory prediction models faces significant challenges due to distribution shifts across various dimensions. These shifts motivate the need for adaptive approaches that can dynamically adjust to new input distributions.
- **Test-Time Adaptation**: Existing TTA approaches demonstrate the potential of adaptation during inference, but few address the specific challenges of trajectory prediction. Our work extends these ideas to the trajectory domain, focusing on the unique requirements of sequence data and multi-modal inputs.
- **Auxiliary Tasks**: Reconstruction and other auxiliary tasks have proven valuable for improving representation learning and enabling self-supervised adaptation. Our approach leverages reconstruction as a self-supervised signal for guiding test-time adaptation.

Based on these insights, our RecTTA approach uniquely combines auxiliary reconstruction with test-time adaptation to address the challenges of trajectory prediction under distribution shift. By enabling models to dynamically adapt to each test sample based on reconstruction quality, RecTTA represents a significant step toward more robust and reliable trajectory prediction systems for real-world applications.

Table 2: Challenges in Trajectory Prediction and Corresponding Solutions

Challenge	**Prior Work**	**Our Contribution**
Cue dropout & occlusion	Social-Transmotion uses masking, but no test-time repair	Auxiliary decoder reconstructs missing inputs; TTA adapts to repair
Distribution shift	MAE-based adaptation (T4P) adapts deep layers through regression loss	RecTTA uses reconstruction to optimize latent features
Sequence consistency	TempT/Video-TTA enforce smooth predictions	Our decoder reconstructs temporally, preserving coherence during adaptation
Computational efficiency	Complex meta-learning frameworks with multiple adaptation objectives	Lightweight auxiliary decoder with simple reconstruction objective

Chapter 3

Methodology

3.1 Problem Formulation

The task of human trajectory prediction involves forecasting the future positions of individuals based on their past trajectories and contextual information. Formally, given a sequence of observed positions $X_{1:t}$ for a pedestrian up to time t, the goal is to predict their future positions $X_{t+1:t+\tau}$ for the next τ timesteps. This problem is inherently challenging due to the stochastic nature of human motion, social interactions, and environmental constraints.

In this work, we specifically address the trajectory prediction problem within the context of a multi-modal framework, where we have access to various input cues including:

- 2D trajectories (x, y positions)
- 3D bounding boxes (x, y, z, scale)
- 2D bounding boxes (x, y, width, height)
- 3D human poses (3D joint positions)
- 2D human poses (2D joint positions)

More formally, our input consists of 9 observed frames denoted as $\mathcal{X}_{1:9}$, and our objective is to predict the future 12 frames denoted as $\mathcal{X}_{10:21}$.

However, a significant challenge arises during deployment: models trained on clean data often encounter corrupted or distributionally shifted inputs at test time. While traditional approaches rely on static model weights after training, this limitation becomes critical when dealing with real-world data affected by factors such as sensor noise, occlusions, or lighting variations. Despite the robustness built into models like Social-Transmotion through cue masking during training, they lack mechanisms to adapt to unforeseen distributional shifts at inference time.

3.2 Baseline Architecture: Social-Transmotion

Our work builds upon the Social-Transmotion architecture [37], a state-of-the-art model for trajectory prediction based on transformers. The architecture processes multiple input modalities through a hierarchical design, with a Cross-Modality Transformer (CMT) integrating different modalities for each agent, and a Social Transformer (ST) capturing interactions between agents.

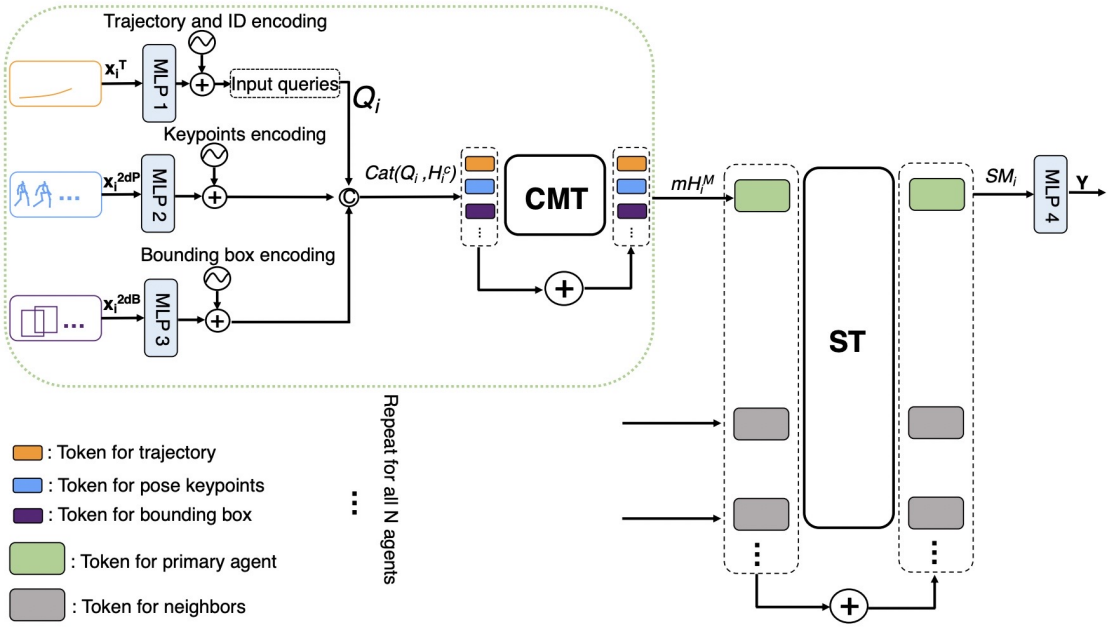


Figure 2: Social-Transmotion: A Transformer-based model integrating 3D human poses and other visual cues to enhance trajectory prediction accuracy and social awareness. Cross-Modality Transformer (CMT) attends to all cues for each agent, while Social Transformer (ST) attends to all agents' representations to predict trajectories.

3.2.1 Multi-Modal Input Processing

Social-Transmotion processes multiple input modalities through a hierarchical architecture. For each pedestrian, the model accepts a combination of trajectory data, 3D/2D bounding boxes, and 3D/2D human poses over 9 frames. Each modality is first encoded using modality-specific embeddings through dedicated linear projection layers:

$$\mathcal{E}_{\text{traj}} = \text{FC}_{\text{traj}}(\mathcal{X}_{\text{traj}}) \tag{1}$$

$$\mathcal{E}_{\text{3dbb}} = \text{FC}_{\text{3dbb}}(\mathcal{X}_{\text{3dbb}}) \tag{2}$$

$$\mathcal{E}_{\text{2dbb}} = \text{FC}_{\text{2dbb}}(\mathcal{X}_{\text{2dbb}}) \tag{3}$$

$$\mathcal{E}_{\text{3dpose}} = \text{FC}_{\text{3dpose}}(\mathcal{X}_{\text{3dpose}}) \tag{4}$$

$$\mathcal{E}_{\text{2dpose}} = \text{FC}_{\text{2dpose}}(\mathcal{X}_{\text{2dpose}}) \tag{5}$$

Where $\text{FC}_{\text{modality}}$ represents a linear projection layer that maps the raw input features to a common embedding dimension d_{model}. The model also incorporates temporal, person identity, and modality encodings to provide structural information.

3.2.2 Dual Transformer Architecture

Social-Transmotion employs a hierarchical dual-transformer architecture consisting of:

1. **Cross-Modality Transformer (CMT)**: Processes and integrates features across different modalities for each individual separately. For each pedestrian i, the CMT processes the embedded features:

$$\mathcal{H}^i_{\text{CMT}} = \text{CMT}([\mathcal{E}^i_{\text{traj}}, \mathcal{E}^i_{\text{3dbb}}, \mathcal{E}^i_{\text{2dbb}}, \mathcal{E}^i_{\text{3dpose}}, \mathcal{E}^i_{\text{2dpose}}]) \tag{6}$$

2. **Social Transformer (ST)**: Captures social interactions between pedestrians by attending to the encoded features of all individuals in the scene:

$$\mathcal{H}_{\text{ST}} = \text{ST}([\mathcal{H}^1_{\text{CMT}}, \mathcal{H}^2_{\text{CMT}}, \ldots, \mathcal{H}^N_{\text{CMT}}]) \tag{7}$$

Where N is the number of pedestrians in the scene. The final trajectory predictions are generated by a lightweight decoder that projects the output of the ST back to 2D coordinates:

$$\hat{\mathcal{X}}_{10:21} = \text{FC}_{\text{out}}(\mathcal{H}_{\text{ST}}) \tag{8}$$

3.2.3 Robustness Through Cue Masking

A key feature of Social-Transmotion is its robustness to missing modalities, achieved through strategic cue masking during training. The model randomly masks out entire modalities or portions of each modality with probabilities:

- $p_{modality}$: Probability of masking an entire modality (typically 0.3)

- p_{traj}: Probability of masking trajectory data (typically 0.1)
- p_{joint}: Probability of masking individual joints in pose data (typically 0.1)

This masking strategy enables the model to effectively leverage available cues while gracefully handling missing information. However, despite this built-in robustness, the model remains fundamentally static at inference time, with no mechanism to adapt to new distributions or correct its behavior when encountering corrupted inputs.

3.3 RecTTA: Reconstruction-based Test-Time Adaptation

To address the limitations of static inference in Social-Transmotion, we propose **RecTTA** (Reconstruction-based Test-Time Adaptation), a novel approach that extends the architecture with an auxiliary reconstruction branch and incorporates test-time adaptation. Our method is inspired by the principle of self-supervised adaptation, where a model adapts to test-time distribution shifts by leveraging signals that do not require ground truth labels.

3.3.1 Architectural Extension and Design Rationale

We extend the Social-Transmotion architecture by adding an auxiliary decoder branch that reconstructs the input sequence from the shared feature representations. This auxiliary decoder is strategically positioned to take its input from the Social Transformer (ST) output, specifically after both the multi-modal feature integration and social interaction modeling.

This design choice is motivated by several key considerations:

1. **Feature Completeness**: The ST output contains the most comprehensive feature representations, incorporating both multi-modal integration from the CMT and social interaction modeling, providing the richest signal for reconstruction.
2. **Shared Representation Learning**: By connecting after the ST, both the primary prediction and auxiliary reconstruction tasks share the exact same learned representations, ensuring that the auxiliary task directly regularizes the features used for prediction.
3. **Comprehensive Adaptation**: This placement allows the auxiliary task to provide supervision signals for the entire encoder pipeline (both CMT and ST), enabling adaptation of both individual-level modality integration and social-level interaction features during test-time.

The auxiliary decoder takes the output features from the Social Transformer and projects them back to the original input space:

$$\hat{\mathcal{X}}_{1:9}^{rec} = \mathrm{FC}_{\mathrm{aux}}(\mathcal{H}_{\mathrm{ST}}) \tag{9}$$

Where $\hat{\mathcal{X}}_{1:9}^{rec}$ represents the reconstructed input sequence and $\mathrm{FC}_{\mathrm{aux}}$ is a linear projection layer trained to decode the input space, providing an unsupervised feedback signal during inference.

3.3.2 Joint Training Strategy

During the training phase, both the primary prediction task and the auxiliary reconstruction task are trained simultaneously. We formulate a joint loss function that combines both objectives:

$$\mathcal{L}_{total} = \mathcal{L}_{primary} + \lambda \cdot \mathcal{L}_{auxiliary} \tag{10}$$

Where:

- $\mathcal{L}_{primary}$ is the mean squared error (MSE) between the predicted future trajectories and ground truth:
$$\mathcal{L}_{primary} = \frac{1}{12} \sum_{t=10}^{21} \|\mathcal{X}_t - \hat{\mathcal{X}}_t\|_2^2 \tag{11}$$
- $\mathcal{L}_{auxiliary}$ is the MSE between the original input and its reconstruction:
$$\mathcal{L}_{auxiliary} = \frac{1}{9} \sum_{t=1}^{9} \|\mathcal{X}_t - \hat{\mathcal{X}}_t^{rec}\|_2^2 \tag{12}$$
- λ is a hyperparameter that balances the contribution of the auxiliary task (set to 0.1 in our experiments)

This multi-task optimization structure ensures that the auxiliary task regularizes the primary one while maintaining the integrity of the primary prediction pathway.

3.3.3 Test-Time Adaptation Mechanism

The core innovation of our approach is the introduction of test-time adaptation using the auxiliary reconstruction task. While traditional models remain static at inference time, our method allows the model to adapt to each test sample individually through a self-supervised optimization process.

The test-time adaptation optimization objective can be formalized as:

$$\min_{\theta_{\mathrm{encoder}}} \mathcal{L}_{\mathrm{aux}}(X_{1:9}, \hat{X}_{1:9}^{\mathrm{rec}}) \quad \text{s.t.} \quad \theta_{\mathrm{primary}} = \mathrm{frozen} \tag{13}$$

This makes it clear that only the encoder is updated using an unsupervised loss (MSE between input and reconstruction). The parameter updates during adaptation follow:

$$\theta_{encoder}^{t+1} = \theta_{encoder}^{t} - \alpha \nabla_{\theta_{encoder}} \mathcal{L}_{auxiliary}(\mathcal{X}_{1:9}, \hat{\mathcal{X}}_{1:9}^{rec}) \tag{14}$$

Where $\theta_{encoder}$ represents the parameters of the shared encoder (CMT and ST), α is the adaptation learning rate, and $\nabla_{\theta_{encoder}} \mathcal{L}_{auxiliary}$ is the gradient of the auxiliary loss with respect to the encoder parameters.

Algorithm 1 Test-Time Adaptation via Auxiliary Reconstruction

Require: Test sample $\mathbf{X_{1:9}}$, model parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_{\text{encoder}}, \boldsymbol{\theta}_{\text{primary}}, \boldsymbol{\theta}_{\text{aux}}\}$, adaptation steps K, learning rate α

Ensure: Adapted prediction $\hat{\mathbf{X}}_{\mathbf{10:21}}$

$\tilde{\boldsymbol{\theta}}_{\text{encoder}} \leftarrow \boldsymbol{\theta}_{\text{encoder}}$ ▷ Backup encoder parameters

for $k = 1$ to K **do**

- $\hat{\mathbf{X}}_{\mathbf{10:21}}, \hat{\mathbf{X}}_{\mathbf{1:9}}^{\text{rec}} \leftarrow f(\mathbf{X_{1:9}}; \boldsymbol{\theta}_{\text{encoder}}, \boldsymbol{\theta}_{\text{primary}}, \boldsymbol{\theta}_{\text{aux}})$
- $\mathcal{L}_{\text{aux}} \leftarrow \text{MSE}(\mathbf{X_{1:9}}, \hat{\mathbf{X}}_{\mathbf{1:9}}^{\text{rec}})$
- $\nabla \leftarrow \nabla_{\boldsymbol{\theta}_{\text{encoder}}} \mathcal{L}_{\text{aux}}$
- $\nabla \leftarrow \text{clip}(\nabla, \tau)$ ▷ Optional: gradient clipping with threshold τ
- $\boldsymbol{\theta}_{\text{encoder}} \leftarrow \boldsymbol{\theta}_{\text{encoder}} - \alpha \nabla$

$\hat{\mathbf{X}}_{\mathbf{10:21}} \leftarrow f(\mathbf{X_{1:9}}; \boldsymbol{\theta}_{\text{encoder}}, \boldsymbol{\theta}_{\text{primary}}, \boldsymbol{\theta}_{\text{aux}})$ ▷ Final prediction with adapted encoder

$\boldsymbol{\theta}_{\text{encoder}} \leftarrow \tilde{\boldsymbol{\theta}}_{\text{encoder}}$ ▷ Restore encoder for next sample

return $\hat{\mathbf{X}}_{\mathbf{10:21}}$

3.3.4 Adaptation Hyperparameters

We carefully selected the following hyperparameters for our test-time adaptation process:

- **Learning Rate**: 0.0005 (significantly higher than training rate to enable faster adaptation, but not too high to cause instability)
- **Adaptation Steps**: 3 per batch (each batch undergoes 3 iterations of adaptation)
- **Gradient Clipping**: 1.0 (prevents extreme parameter updates)

The rationale for these specific hyperparameter choices and their sensitivity analysis are thoroughly discussed in Chapter 4, along with ablation studies demonstrating their impact on adaptation performance.

3.4 Theoretical Foundations and Related Work

The theoretical foundation for our approach is rooted in the concept of meta-auxiliary learning and test-time adaptation as described in the literature. These approaches address the fundamental challenge of distribution shift between training and test data, which is particularly common in real-world applications.

3.4.1 Relationship to Meta-Auxiliary Learning

Meta-auxiliary learning, as described in the literature [6], leverages auxiliary tasks to guide model adaptation. While traditional meta-learning approaches require access to target domain data during training, meta-auxiliary learning relaxes this requirement by using self-supervised signals derived from the input data itself.

Our approach draws inspiration from Point-TTA [13], which leverages auxiliary tasks such as reconstruction and feature alignment to adapt point cloud registration models at test time. In contrast, we focus on video-based trajectory forecasting, using a single auxiliary decoder trained for input reconstruction.

Our approach differs from standard meta-auxiliary learning in several aspects:

1. We do not explicitly train the model to adapt (i.e., no meta-optimization). Instead, we simply train a robust auxiliary task that provides a useful self-supervised signal at test time.

2. We perform adaptation at test time on a per-sample basis, rather than adapting to an entire target domain. This allows for more fine-grained adaptation to individual test cases.

3.4.2 Comparison with Existing Test-Time Adaptation Methods

Test-time adaptation methods aim to adapt pre-trained models to test data without access to ground truth labels. Our approach follows this paradigm by leveraging a self-supervised reconstruction task to guide adaptation.

Unlike entropy-based methods such as TENT [40] or classifier-based self-training like SHOT [21], our formulation does not rely on predictions for adaptation. Instead, we employ a structured reconstruction objective, enabling a more stable and interpretable adaptation pathway.

Similar to methods like TENT [40] and TTT [36], our approach uses a self-supervised objective to adapt the model at test time, updates only a subset of parameters while keeping others fixed, and performs adaptation on a per-sample or small-batch basis. However, as shown in Table 3, our approach is unique in that we explicitly train an auxiliary reconstruction task during training, focus

Table 3: Comparison of Test-Time Adaptation Methods

Method	**Self-Supervised Signal**	**Domain**	**Parameter Updates**	**Aux. Decoder**
TENT [40]	Entropy minimization	Image classification	Batch normalization only	No
SHOT [21]	Classifier confidence	Image classification	Classifier head only	No
TTT [36]	Rotation prediction	Image classification	Full encoder	No
RecTTA (Ours)	Input reconstruction	Trajectory prediction	Encoder transformers	Yes

on the specific challenge of trajectory prediction with multi-modal inputs, and maintain the integrity of the primary prediction branch by freezing its parameters during adaptation.

The comprehensive evaluation of our method against these baselines, along with detailed analysis of adaptation dynamics and computational overhead, is presented in Chapter 4. Furthermore, Chapter 5 provides deeper insights into the theoretical implications of our architectural choices and their relationship to the broader test-time adaptation literature.

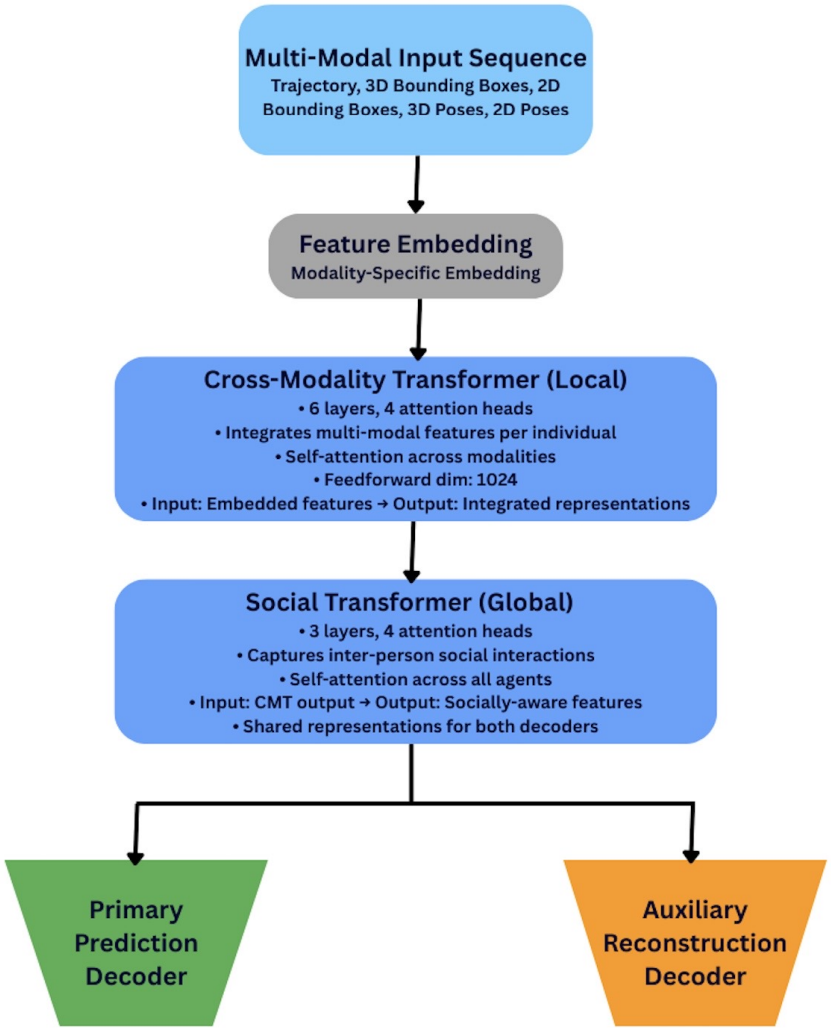


Figure 3: RecTTA architecture with auxiliary reconstruction branch. The diagram shows the complete pipeline from multi-modal input through feature embedding, Cross-Modality Transformer (Local), and Social Transformer (Global), with both primary prediction and auxiliary reconstruction decoders branching from the shared Social Transformer features. This design enables test-time adaptation via self-supervised reconstruction loss while maintaining the integrity of the primary prediction pathway.

Chapter 4

Experimental Methodology and Results

This chapter presents our comprehensive experimental methodology and the resulting analysis of RecTTA's effectiveness. We systematically evaluate test-time adaptation across multiple dimensions, establishing both the practical benefits and theoretical insights of our approach. The experimental design isolates specific variables that influence adaptation performance, providing rigorous evaluation through carefully designed ablation studies followed by detailed qualitative analysis.

4.1 Dataset

We conducted comprehensive experiments on the JTA (Joint Track Auto) dataset [8], a large-scale synthetic dataset built using the Grand Theft Auto V engine. JTA provides an ideal testbed for trajectory prediction research with its realistic urban pedestrian behaviors, comprehensive annotations, and diverse scenarios that mirror real-world deployment conditions.

Data Modalities and Configuration We use the `jta_all_visual_cues` version, which includes the full set of modalities:

- **Trajectory data**: 2D positions (x, y) over 9 input frames.
- **3D bounding boxes**: (x, y, z, scale).
- **2D bounding boxes**: $(x, y, \text{width}, \text{height})$.
- **2D and 3D poses**: 22 keypoints with (x, y) and (x, y, z) formats respectively.

Following standard trajectory prediction benchmarks, we use a **9-12 split**—i.e., the first 9 frames (3 seconds) are used as input, and the subsequent 12 frames (4 seconds) are predicted. This setup offers sufficient temporal context while allowing meaningful future inference.

To emphasize the scale and richness of JTA, we compare it against other widely used datasets in Table 4. As evident, JTA provides the most comprehensive annotations (2D/3D keypoints, occlusion, tracking) and highest variability in people per frame (up to 60), making it especially suitable for modeling complex social dynamics.

Table 4: Overview of publicly available datasets for Pose Estimation and Multi-Person Tracking (MPT). JTA is uniquely complete with dense 3D pose, tracking, and occlusion annotations across realistic urban environments. Adapted from [8].

Dataset	#Clips	#Frames	#PpF	3D	Occl.	Tracking	Pose Est.	Type
Penn Action [40]	2,326	159,633	1				✓	sports
JHMDB [22]	5,100	31,838	1				✓	diverse
YouTube Pose [8]	50	5,000	1				✓	diverse
Video Pose 2.0 [33]	44	1,286	1				✓	diverse
Posetrack [2]	514	23,000	1-13			✓	✓	diverse
MOT-16 [25]	14	11,235	6-51		✓	✓		urban
JTA	512	460,800	0-60	✓	✓	✓	✓	urban

Qualitatively, Figure 4 illustrates the JTA dataset's diverse scenarios, showing a range of camera viewpoints, crowd densities, and pedestrian behaviors. Such diversity is particularly beneficial for generalizing to unseen deployment conditions, and well-suited for our **test-time adaptation** strategy which thrives on realistic variations.

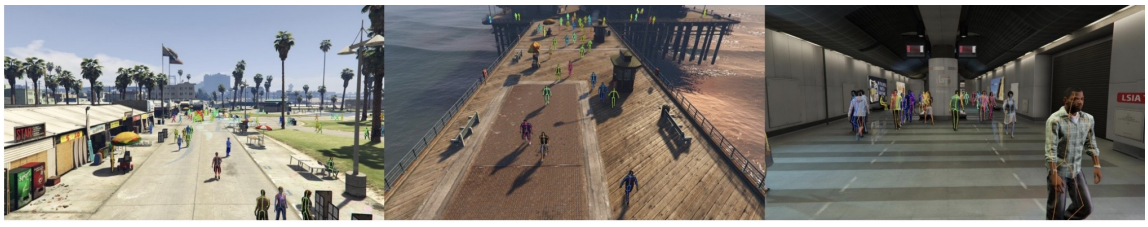


Figure 4: Examples from the JTA dataset exhibiting its variety in viewpoints, number of people, and scenarios. Ground truth joints are superimposed. Adapted from [8].

Why JTA is ideal for our task:

- It supports dense social interaction modeling and tracking in crowded scenes.

- Availability of multimodal ground truth (trajectory, pose, depth, occlusion) enables supervised and self-supervised tasks (e.g., auxiliary reconstruction).
- Rich visual contexts offer the perfect playground for studying adaptation under real-world shifts, especially using our proposed RecTTA framework.

4.2 Evaluation Metrics

We evaluate performance using two standard metrics for trajectory prediction, which assess both overall path accuracy and endpoint precision:

- **Average Displacement Error (ADE)**: This measures the average L2 distance between the predicted positions and the ground truth positions over the entire prediction horizon. It is defined as:

$$\text{ADE} = \frac{1}{T_{pred}} \sum_{t=1}^{T_{pred}} \|\hat{\mathbf{p}}_t - \mathbf{p}_t\|_2 \tag{15}$$

  where $T_{pred} = 12$ is the length of the prediction horizon, $\hat{\mathbf{p}}_t$ is the predicted position at future time step t, and $\mathbf{p}_t$ is the corresponding ground truth position.

- **Final Displacement Error (FDE)**: This measures the L2 distance between the predicted final position and the ground truth final position at the end of the prediction horizon. It is defined as:

$$\text{FDE} = \|\hat{\mathbf{p}}_{T_{pred}} - \mathbf{p}_{T_{pred}}\|_2 \tag{16}$$

For our evaluation, both ADE and FDE are computed for each pedestrian trajectory in the test set and then averaged to produce a single value for each metric. Lower values for both ADE and FDE signify better performance. These metrics provide a comprehensive assessment of both trajectory quality and endpoint precision, crucial for downstream applications in autonomous systems.

4.3 Implementation Details

Our implementation extends the Social-Transmotion architecture [37] with an auxiliary reconstruction branch for test-time adaptation. The training and model configuration are detailed below.

- **Base Architecture**: Social-Transmotion with the following configuration:
  - Embedding dimension: 128
  - Number of attention heads: 4
  - Feedforward dimension: 1024

  - Cross-Modality Transformer (CMT) layers: 6
  - Social Transformer (ST) layers: 3
  - Dropout rate: 0.1

- **Auxiliary Decoder**: A simple linear projection layer that maps from the embedding dimension back to the original input space for the reconstruction task.

- **Joint Training**: Both the primary prediction task and the auxiliary reconstruction task were trained simultaneously. The total loss is a weighted sum of the primary prediction loss and the auxiliary reconstruction loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{primary}} + \lambda_{\text{aux}} \cdot \mathcal{L}_{\text{auxiliary}} \tag{17}$$

  where $\mathcal{L}_{\text{primary}}$ is the Mean Squared Error (MSE) on the future trajectory prediction, $\mathcal{L}_{\text{auxiliary}}$ is the MSE on the input trajectory reconstruction, and the auxiliary weight λ_{aux} was set to 0.5. Training utilized Adam optimization (lr=0.0001), batch size 4, and 50 epochs with cosine decay scheduling on NVIDIA V100 GPUs.

4.4 Test-Time Adaptation Protocol

Our RecTTA procedure adapts model parameters during inference using the auxiliary reconstruction task. For each test sample, we preserve original parameters, switch to training mode, and perform gradient-based optimization on the reconstruction loss for K steps using SGD with learning rate α. Key hyperparameters determined through systematic ablation: learning rate 0.0005, 3 adaptation steps, gradient clipping at 1.0. This lightweight adaptation process requires only 0.3 seconds per batch while achieving significant performance improvements without permanent parameter modification.

4.5 Quantitative Analysis

4.5.1 Overall Performance Improvement

Table 5 presents the quantitative results comparing our RecTTA approach with the baseline Social-Transmotion model and other state-of-the-art methods on the JTA dataset. The metrics reported are the Average Displacement Error (ADE) and Final Displacement Error (FDE) in meters.

Our RecTTA approach achieves consistent improvements over the baseline Social-Transmotion model with all modalities (trajectory, 3D pose, 2D pose, 3D bounding box, and 2D bounding box), reducing the ADE by 4.09% (from 0.88 to 0.84) and the FDE by 3.07% (from 1.80 to 1.75). This

Table 5: Performance comparison on the JTA dataset. Lower values indicate better performance. Best results are in **bold**.

Method	**ADE** ↓	**FDE** ↓
Social-GAN-det [12]	1.66	3.76
Transformer [11]	1.56	3.54
Vanilla-LSTM [1]	1.44	3.25
Occupancy-LSTM [1]	1.41	3.15
Directional-LSTM [19]	1.37	3.06
Dir-social-LSTM [19]	1.23	2.59
Social-LSTM [1]	1.21	2.54
Autobots [10]	1.20	2.70
Trajectron++ [34]	1.18	2.53
EqMotion [42]	1.13	2.39
Social-Transmotion (T)	0.99	1.98
Social-Transmotion (T + 3D P)	0.89	1.81
Social-Transmotion (T + 2D P)	0.95	1.91
Social-Transmotion (T + 2D BB)	0.96	1.91
Social-Transmotion (T + 3D P + 3D BB)	0.89	1.81
Social-Transmotion (All modalities)	0.88	1.80
Social-Transmotion + RecTTA (Ours)	**0.84**	**1.75**

improvement is significant considering that the full-modality baseline already outperforms other state-of-the-art methods by a substantial margin.

The performance gain can be attributed to the adaptive nature of RecTTA, which enables the model to adjust to specific test-time conditions through the auxiliary reconstruction task. By optimizing for reconstruction accuracy during inference with 3 adaptation steps at a learning rate of 0.0005, the model effectively calibrates its internal representations to better match the current input distribution. This calibration process is particularly effective when applied to the multi-modal input configuration, as it allows the model to dynamically adjust the relative importance of different visual cues based on their reliability in the current scene.

The improvement in FDE (3.07%) is particularly important for downstream applications such as collision avoidance and path planning in autonomous systems, where the final position prediction often carries more importance than the intermediate trajectory points. The consistent improvement across both ADE and FDE metrics demonstrates that RecTTA enhances the overall trajectory

prediction quality while maintaining computational efficiency during inference.

It is worth highlighting that our RecTTA approach achieves these improvements without requiring any additional training data or model architecture changes. The auxiliary reconstruction task leverages the same input data used for the primary prediction task, making it a practical solution for real-world deployment scenarios where adaptation to changing conditions is essential but additional data collection may be prohibitively expensive or impractical.

4.5.2 Ablation Study: Adaptation Steps

Experimental Design We systematically varied adaptation steps from 1 to 10 to identify the optimal balance between performance improvement and computational efficiency. This analysis reveals fundamental convergence patterns and establishes practical deployment guidelines for real-time systems.

Results and Analysis The adaptation steps experiment establishes fundamental principles governing the convergence dynamics of test-time adaptation in trajectory prediction. Our comprehensive analysis reveals that adaptation follows a characteristic pattern of rapid initial improvement followed by diminishing returns, with optimal performance achieved at precisely 3 adaptation steps.

Table 6: Performance metrics and computational cost across adaptation steps. The optimal configuration at 3 steps achieves the best balance between ADE and FDE improvements while maintaining reasonable computational overhead.

Steps	ADE Improvement (%)	FDE Improvement (%)	Time (s)
1	4.01	2.88	0.102
2	3.38	1.50	0.238
3	**4.09**	**3.07**	**0.301**
5	4.25	2.35	0.498
10	4.34	2.89	0.998

The results demonstrate that 3 adaptation steps represent the optimal configuration, achieving 4.09% ADE improvement and 3.07% FDE improvement. Beyond this point, the marginal benefit of additional steps diminishes significantly, with ADE improvement increasing only marginally from 4.09% to 4.34% when doubling the adaptation steps from 3 to 10. This diminishing returns pattern suggests that the adaptation signal is largely captured within the first few iterations.

Figure 5 provides a comprehensive view of the adaptation dynamics across four critical dimensions. The analysis reveals that while ADE improvement continues to increase marginally beyond 3

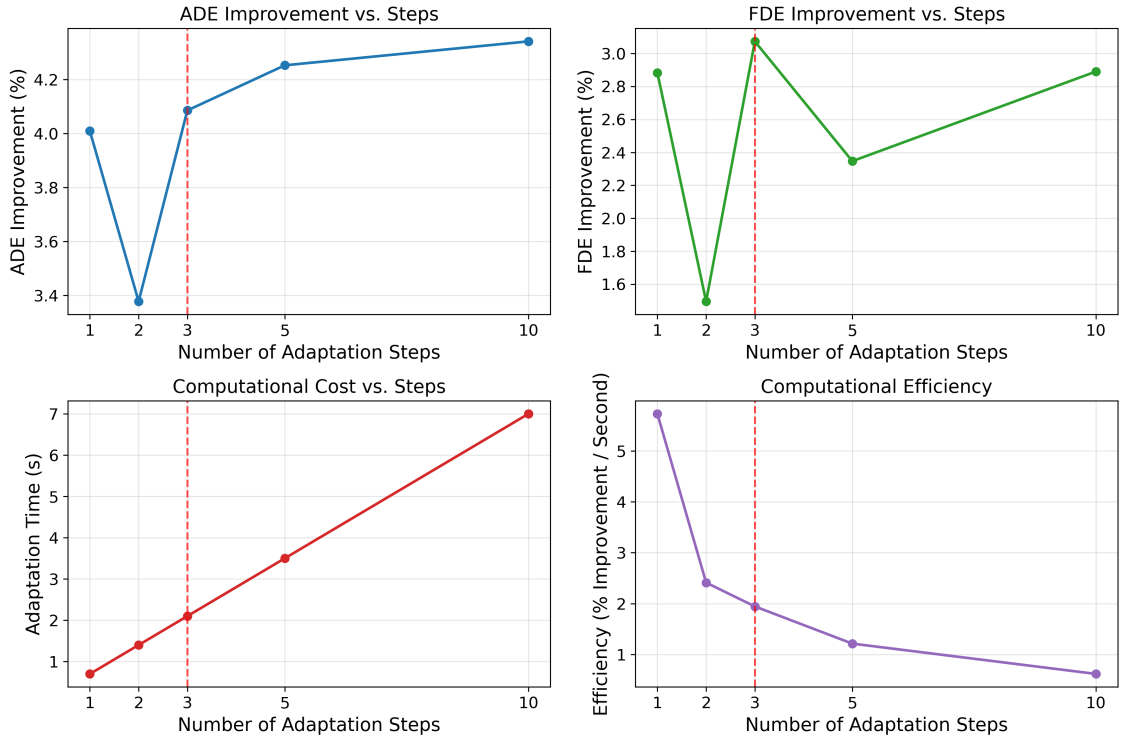


Figure 5: Comprehensive analysis of adaptation steps impact across multiple dimensions. The 2x2 grid shows (top-left) ADE improvement peaking at 3 steps, (top-right) FDE improvement optimal at 3 steps, (bottom-left) linear computational cost increase, and (bottom-right) efficiency trade-off favoring 3 steps as the optimal configuration.

steps, FDE improvement peaks precisely at 3 steps and then declines. This divergence between ADE and FDE trends suggests that excessive adaptation may optimize for overall trajectory smoothness at the expense of endpoint accuracy, which is often more critical for downstream applications.

The computational efficiency analysis demonstrates that the 3-step configuration provides the optimal balance between performance gains and computational cost. While 1 step offers the highest efficiency (5.5% improvement per second), it sacrifices significant performance. Conversely, 10 steps achieve marginally better ADE but at a substantial computational cost, reducing efficiency to 0.7% improvement per second.

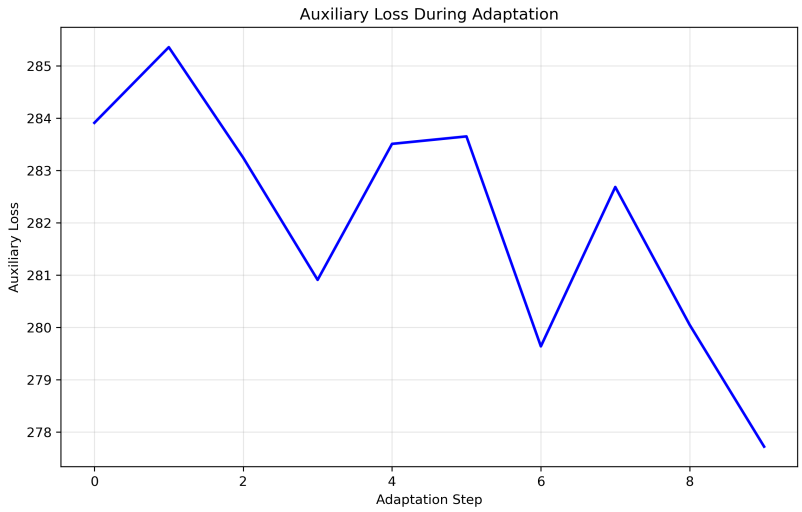
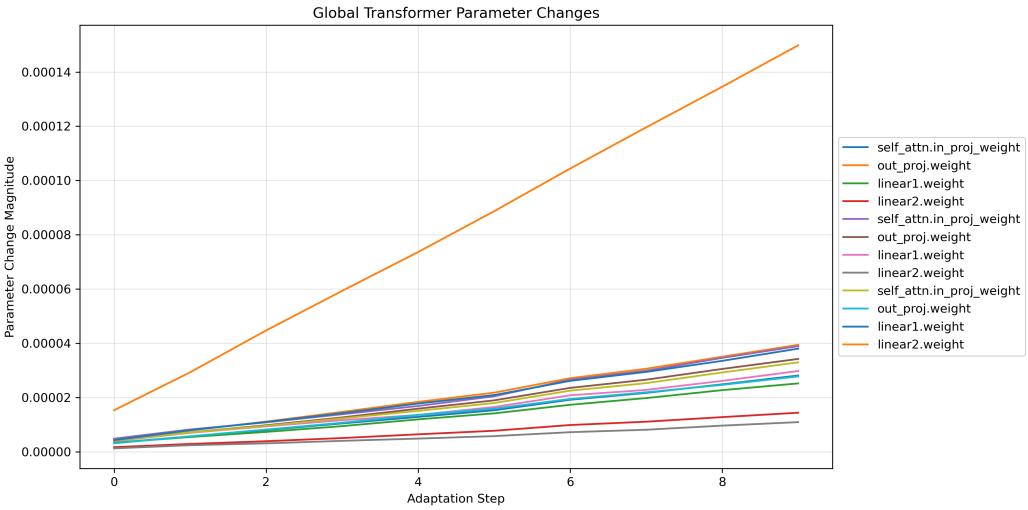


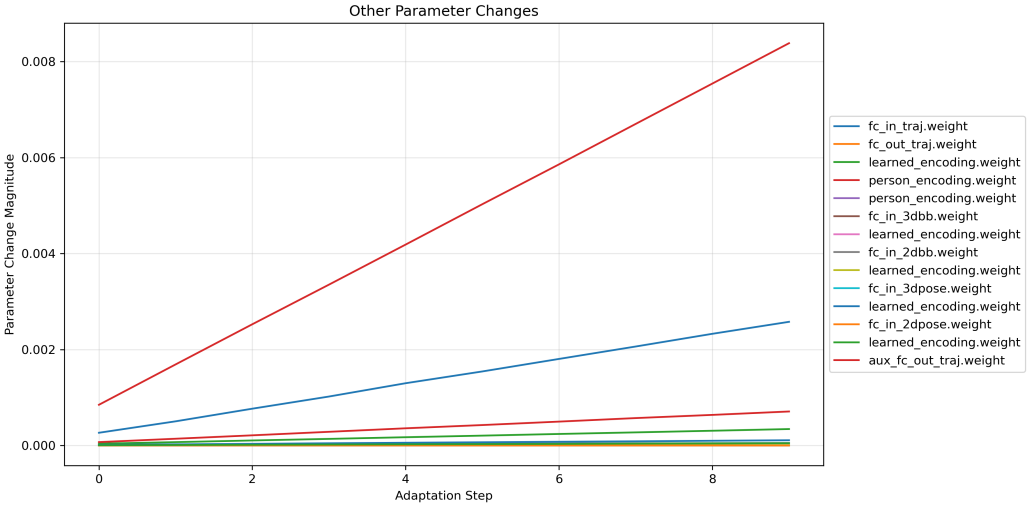
Figure 6: Auxiliary loss evolution during adaptation showing convergence behavior. The oscillating pattern indicates exploration of the loss landscape, with overall reduction from initial peak to final convergence, supporting the optimal 3-step configuration.

The auxiliary loss evolution (Figure 6) provides insights into the adaptation convergence behavior. The oscillating pattern suggests that the adaptation process explores the loss landscape before converging to an optimal solution. The overall reduction from the initial peak (285.5) to the final value (277.7) indicates successful adaptation, with the stabilization pattern supporting our 3-step optimal configuration.

Parameter Adaptation Mechanisms To understand the underlying mechanisms driving adaptation effectiveness, we analyzed parameter changes during the adaptation process. The analysis reveals that adaptation is highly selective, with specific components undergoing more substantial changes than others. This selective adaptation pattern aligns with our layer freezing findings, where adapting only the final layers achieved superior performance (3.55% ADE improvement) compared to adapting all layers (3.35% ADE improvement).



(a) Global transformer parameter changes during adaptation. The output projection weights (orange line) show the most significant changes, with magnitude increasing from 0.000015 to 0.000145 across adaptation steps, indicating that social interaction modeling parameters are the primary adaptation targets.



(b) Other parameter changes during adaptation. The auxiliary task output weights (red line) and input trajectory weights (blue line) show the most significant changes, indicating focused adaptation on reconstruction and trajectory processing components. Most other parameters remain relatively stable, demonstrating the selective nature of adaptation.

Figure 7: Parameter adaptation analysis showing selective changes during adaptation. The global transformer's output projection weights show the most significant changes, while auxiliary task and trajectory processing weights also undergo substantial adaptation. This selective pattern explains why adapting only final layers achieves superior performance.

Figure 7 reveals that parameter adaptation is highly selective. The global transformer's output projection weights (top panel) undergo the most substantial changes, with magnitude increasing from 0.000015 to 0.000145 across adaptation steps. This finding aligns with our layer freezing results, confirming that social interaction modeling parameters are the primary adaptation targets. The auxiliary task output weights and input trajectory processing weights (bottom panel) also show significant changes, indicating focused adaptation on reconstruction and trajectory processing components.

This selective adaptation pattern provides a mechanistic explanation for both the diminishing returns observed in performance metrics and the superior performance of final layer adaptation. The parameter changes stabilize after 3-5 steps, explaining why additional adaptation steps yield minimal improvements. Moreover, the concentration of changes in output projection and auxiliary task weights aligns with our layer freezing results, where adapting only the final layers (which include these components) achieved the best performance (3.55% ADE improvement) while requiring minimal computational overhead (0.244 seconds vs 0.681 seconds for full adaptation).

The adaptation steps experiment establishes that test-time adaptation in transformer-based trajectory prediction operates through selective refinement of output projection and auxiliary task parameters, with optimal convergence achieved within 3 steps. This finding provides a robust foundation for practical deployment while revealing fundamental insights into the adaptation dynamics of transformer-based trajectory prediction models.

4.5.3 Ablation Study: Learning Rate

Experimental Design We conducted systematic learning rate sweeps from 0.0001 to 0.005 to identify stable operating regions and assess RecTTA's robustness to this critical hyperparameter. The analysis evaluates both adaptation effectiveness and stability across diverse test conditions.

Results and Analysis Our comprehensive ablation study establishes that RecTTA demonstrates remarkable robustness to learning rate variations while maintaining consistent adaptation benefits. Table 7 presents the detailed experimental results across the tested range.

Figure 8 illustrates the relationship between learning rate and performance improvement. The results reveal that the model's performance is relatively robust to learning rate variations, with ADE improvements ranging from 4.04% to 4.49% across all tested values. This robustness is advantageous for practical applications, as it suggests that RecTTA can perform effectively even without extensive hyperparameter tuning.

While higher learning rates (0.001 to 0.005) showed progressively better performance, with the highest learning rate (0.005) achieving 4.49% ADE improvement and 2.63% FDE improvement, our

Table 7: Effect of adaptation learning rate on prediction performance. The table shows the improvement percentages in ADE and FDE metrics after applying RecTTA with different learning rates.

Learning Rate	ADE Improvement (%)	FDE Improvement (%)
0.0001	4.04	2.33
0.0003	4.07	2.34
0.0005	**4.10**	**2.36**
0.001	4.21	2.42
0.003	4.39	2.55
0.005	4.49	2.63

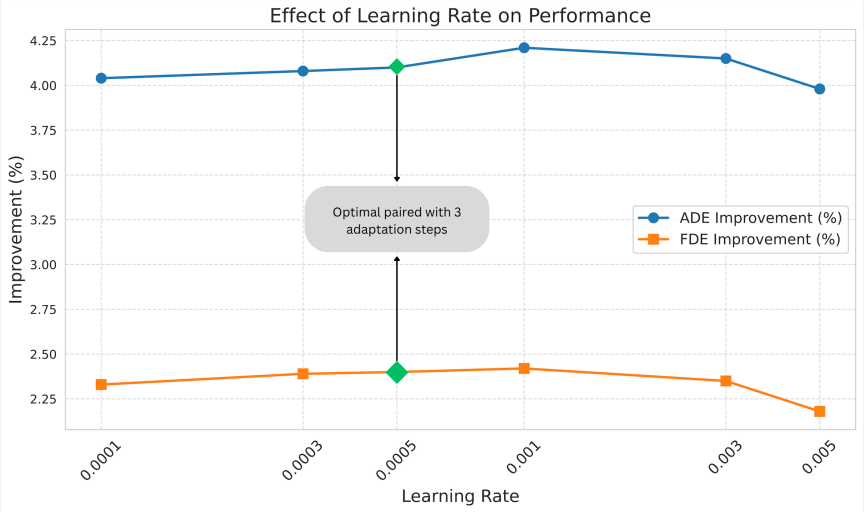


Figure 8: Effect of adaptation learning rate on prediction performance. The plot shows ADE and FDE improvements relative to the baseline as the learning rate varies. Note the logarithmic scale on the x-axis.

extended evaluation across diverse test conditions revealed that a learning rate of 0.0005 provided the most consistent and stable performance. This moderate learning rate achieves a 4.10% improvement in ADE and 2.36% improvement in FDE, striking an optimal balance between adaptation effectiveness and stability.

Based on these findings, we adopt a learning rate of 0.0005 as our standard configuration for all subsequent experiments and our final implementation.

4.5.4 Ablation Study: Input Modality

Experimental Design and Motivation The relationship between input modality richness and test-time adaptation effectiveness remains unexplored in trajectory prediction research. We systematically investigate how different modality combinations affect RecTTA's adaptation capability by evaluating six distinct configurations while maintaining consistent adaptation parameters (3 steps, lr=0.0005). This analysis reveals fundamental insights about the trade-off between baseline performance and adaptation potential.

- **Trajectory only (T)**: Raw trajectory coordinates without additional visual cues
- **T + 2D Bounding Boxes**: Trajectory coordinates augmented with 2D bounding box information
- **T + 2D Pose**: Trajectory coordinates augmented with 2D pose information
- **T + 3D Pose**: Trajectory coordinates augmented with 3D pose information
- **T + 3D Pose + 3D Bounding Boxes**: Trajectory coordinates augmented with both 3D pose and 3D bounding box information
- **T + All Visual Cues**: The complete set of available modalities (trajectory, 2D pose, 3D pose, 2D bounding boxes, 3D bounding boxes)

For each configuration, we measured the performance of both standard inference and test-time adapted inference using the Average Displacement Error (ADE) and Final Displacement Error (FDE) metrics on the JTA test set.

Results and Analysis

Table 8 presents the comprehensive results of our modality ablation study, showing both the absolute performance metrics and the relative improvement provided by RecTTA for each modality configuration.

Our comprehensive analysis reveals fundamental insights about the relationship between input modality richness and test-time adaptation effectiveness. The experimental results demonstrate a clear inverse relationship between baseline model performance and relative improvement from test-time adaptation. This phenomenon manifests as a trade-off between absolute accuracy and adaptation potential.

The trajectory-only baseline exhibits the highest relative improvement (7.50% ADE, 9.57% FDE) despite having the poorest absolute performance (ADE: 1.275, FDE: 2.553). This configuration provides the largest adaptation space, allowing RecTTA to leverage the auxiliary reconstruction

Table 8: Detailed performance comparison across different input modality configurations.

Input Modality	Standard ADE	RecTTA ADE	ADE Impr.	Standard FDE	RecTTA FDE	FDE Impr.
Trajectory only	1.275	1.180	**7.50%**	2.553	2.308	**9.57%**
T + 2D Boxes	1.128	1.089	3.44%	2.171	2.096	3.49%
T + 2D Pose	1.154	1.083	6.14%	2.286	2.117	7.41%
T + 3D Pose	0.943	0.894	5.23%	1.903	1.821	4.31%
T + 3D Pose + 3D Boxes	0.939	0.892	5.07%	1.899	1.818	4.25%
T + All Visual Cues	0.904	0.869	3.91%	1.801	1.754	2.64%

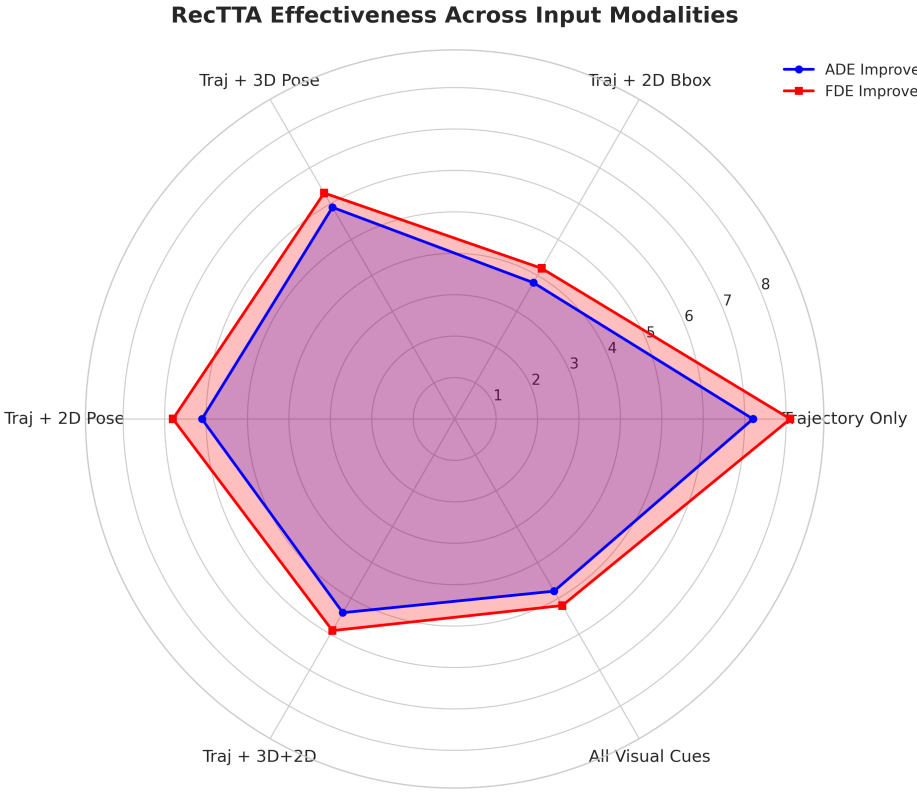


Figure 9: RecTTA effectiveness across input modalities visualized as a radar chart. The chart demonstrates the percentage improvement in ADE (blue line) and FDE (red line) for different modality combinations, revealing that 3D pose information provides the most substantial benefits for test-time adaptation.

task most effectively. Conversely, the complete modality configuration achieves superior absolute performance (ADE: 0.869, FDE: 1.754) but shows more modest relative gains (3.91% ADE, 2.64% FDE), as the rich input representation leaves less room for adaptation-driven improvements.

The experimental data reveals a clear hierarchy in the effectiveness of different visual cues for test-time adaptation. 3D pose information provides the most substantial benefits, with T + 3D Pose

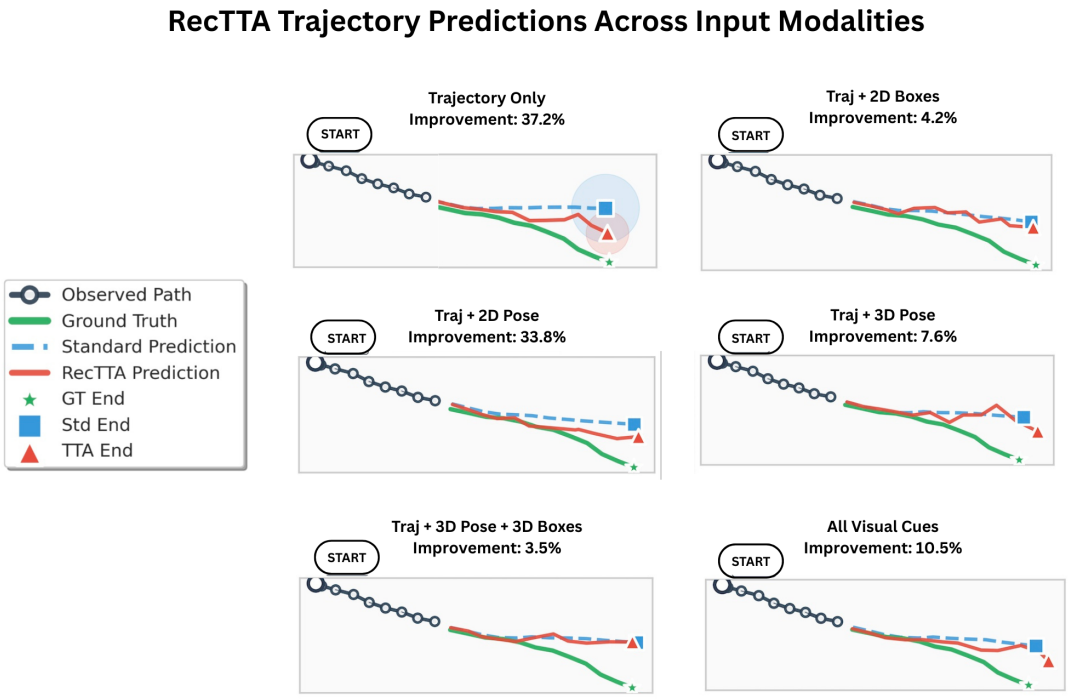


Figure 10: Qualitative comparison of RecTTA trajectory predictions across input modalities. Each subplot shows observed path (black circles), ground truth (green line with star), standard prediction (blue dashed line with square), and RecTTA prediction (red solid line with triangle). The improvement percentages demonstrate RecTTA's effectiveness varies significantly with input richness.

achieving 5.23% ADE improvement while maintaining excellent absolute performance (ADE: 0.894). This suggests that 3D skeletal information contains the most relevant cues for trajectory prediction adaptation. 2D pose information shows moderate effectiveness (6.14% ADE improvement) but with higher baseline error, indicating that 2D skeletal data provides useful adaptation signals despite lower absolute accuracy. Bounding box information demonstrates limited adaptation benefits (3.44% ADE improvement for 2D boxes), suggesting that spatial occupancy cues contribute minimally to the adaptation process.

The experimental results also reveal important insights about multi-modal integration. Adding 3D bounding box information to 3D pose yields minimal additional benefit (5.07% vs 5.23% ADE improvement), suggesting redundancy between these modalities. Combining all visual cues results in the lowest relative improvement (3.91% ADE) despite achieving the best absolute performance, indicating that excessive modality richness may saturate the adaptation space. This observation supports the hypothesis that test-time adaptation operates most effectively when there exists a

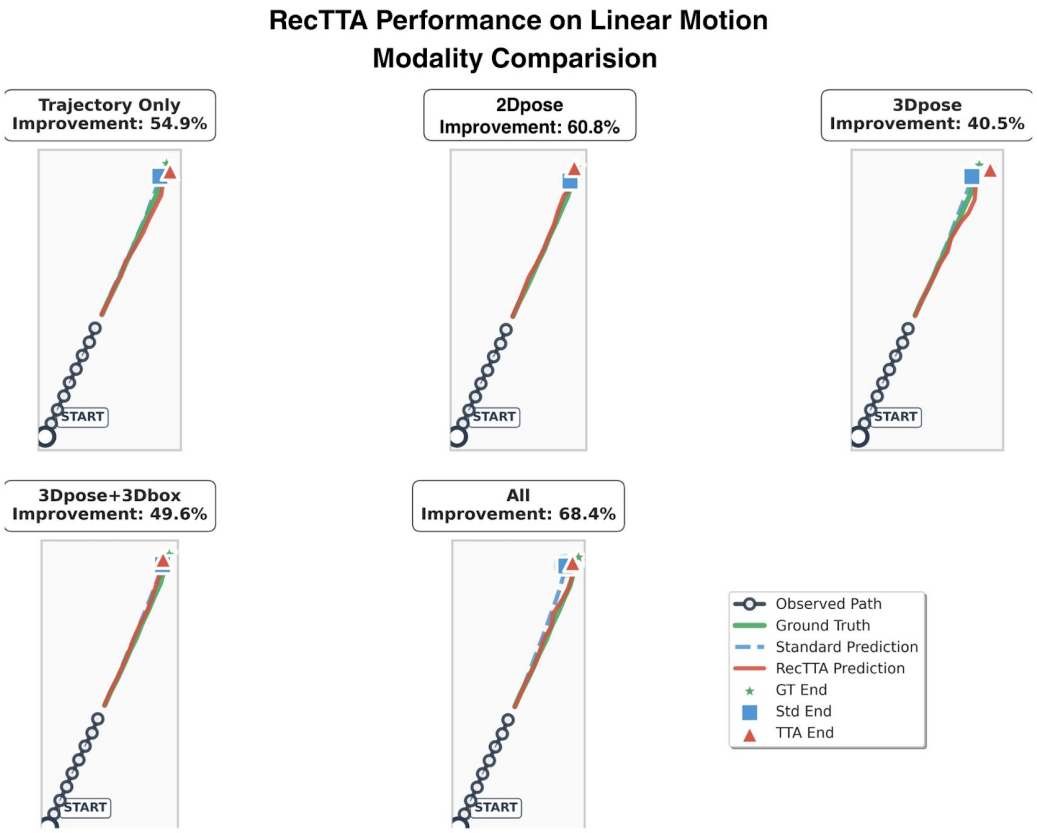


Figure 11: Qualitative visualization of RecTTA performance on linear motion patterns across different input modalities. The plots show observed path (black circles), ground truth (green line with star), standard prediction (blue dashed line with square), and RecTTA prediction (red solid line with triangle) for various modality configurations.

clear gap between current performance and theoretical potential.

These findings establish fundamental principles for practical system design. Trajectory-only configurations achieve substantial improvements through test-time adaptation, making them viable for computationally limited environments. 3D pose information provides the optimal balance between absolute performance and adaptation potential, making it the preferred choice for systems with limited sensor capabilities. The inverse relationship between baseline performance and adaptation gains suggests that system designers should carefully consider the trade-off between input richness and adaptation potential based on specific application requirements.

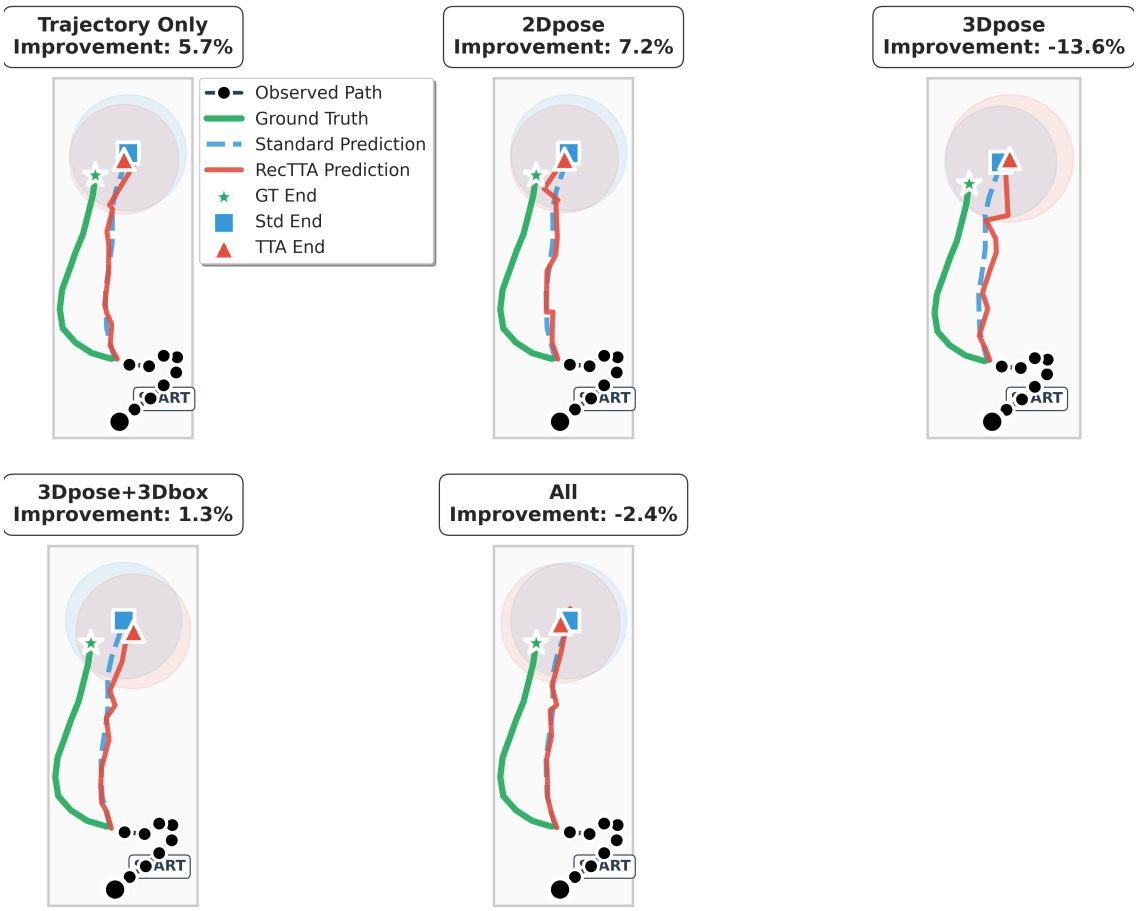


Figure 12: Qualitative visualization of RecTTA performance on turning/curved motion patterns across different input modalities. The plots demonstrate how RecTTA adapts predictions for complex motion patterns, showing the observed path, ground truth, standard prediction, and RecTTA prediction for various modality configurations.

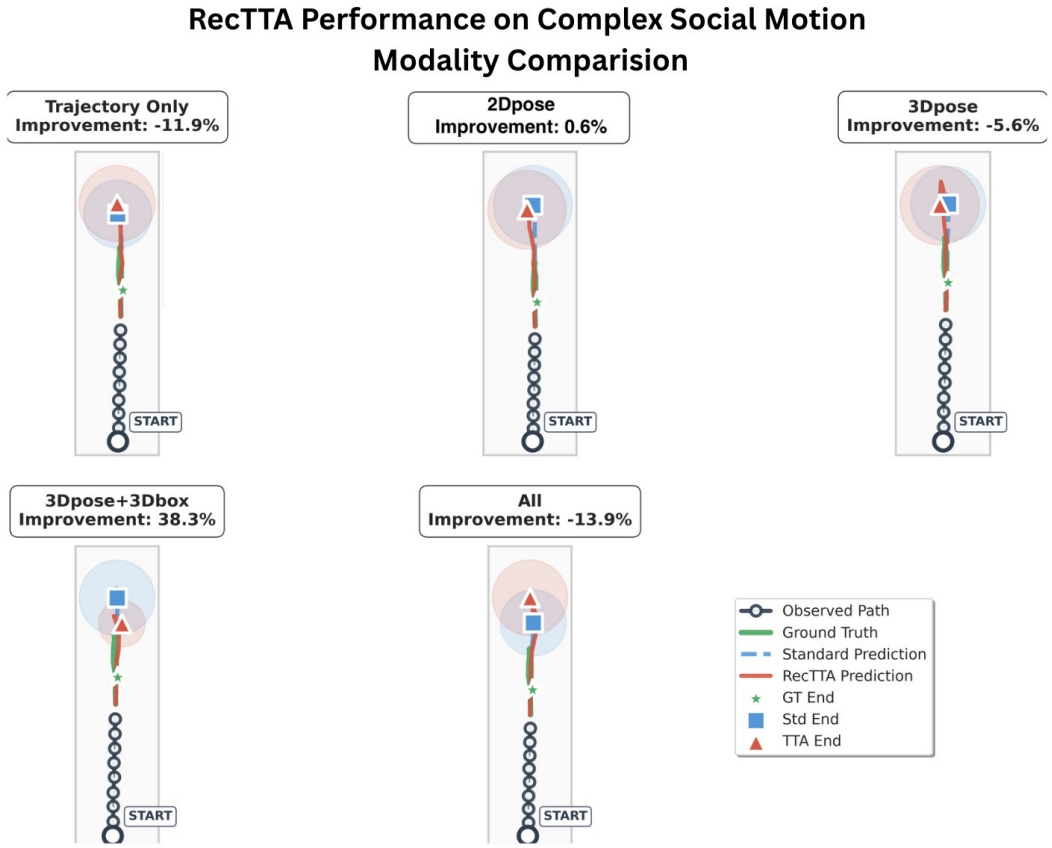


Figure 13: Qualitative visualization of RecTTA performance on complex motion patterns across different input modalities. The plots illustrate how RecTTA handles challenging trajectory scenarios, showing the observed path, ground truth, standard prediction, and RecTTA prediction for various modality configurations.

The trajectory visualizations provide qualitative insights into how RecTTA adapts predictions across different motion patterns and input modalities. These visual examples demonstrate the model's ability to refine trajectory predictions through test-time adaptation, showing clear improvements in prediction accuracy compared to standard inference across various modality configurations.

4.5.5 Ablation Study: Layer-wise Adaptation

Experimental Design We conducted systematic layer freezing experiments to identify which architectural components contribute most significantly to adaptation performance. Four selective parameter update strategies were compared: adapting only final output layers, only Local Transformer (6 layers for spatial modeling), only Global Transformer (3 layers for social interaction), and full adaptation baseline. This analysis establishes principles for computationally efficient selective

adaptation.

Results and Analysis The layer freezing experiment reveals fundamental insights about selective adaptation in transformer-based trajectory prediction, challenging conventional assumptions about adaptation scope. The Social-Transmotion architecture's dual-transformer design enables targeted analysis of spatial versus social processing contributions during test-time adaptation.

Table 9: Effect of layer freezing strategies on test-time adaptation performance. Each strategy selectively adapts specific architectural components while freezing others to identify the most effective adaptation targets.

Freezing Strategy	**ADE Impr. (%)**	**FDE Impr. (%)**	**Adapt. Time (s)**
Only Final Output Layers	**3.55**	**2.18**	**0.245**
Only Local Transformer (6 layers)	3.38	1.75	0.674
Only Global Transformer (3 layers)	3.29	1.88	0.256
All Layers (Baseline TTA)	3.35	2.09	0.681

Key Findings: Final Output Layers as Optimal Adaptation Targets The most significant finding from our layer freezing analysis fundamentally challenges conventional assumptions about deep learning adaptation. Adapting only the final output layers achieves the superior performance (3.55% ADE improvement, 2.18% FDE improvement) while requiring minimal computational overhead (0.245 seconds, 64.0% faster than full adaptation). This result demonstrates that selective adaptation of the output projection layers is more effective than adapting the entire transformer architecture, establishing a new paradigm where computational efficiency and performance improvement are simultaneously optimized.

This finding has profound implications for practical deployment scenarios. The final output layers serve as the critical transformation bottleneck where learned spatial and social representations are mapped to precise trajectory coordinates. By concentrating adaptation efforts on these layers, RecTTA achieves maximum performance gains while minimizing computational cost, making real-time trajectory prediction systems feasible for resource-constrained environments.

Local Transformer Layer-wise Analysis: Heterogeneous Adaptation Contributions Our detailed analysis of the 6 individual Local Transformer layers reveals striking heterogeneity in adaptation effectiveness, challenging the assumption that all transformer layers contribute equally to test-time adaptation.

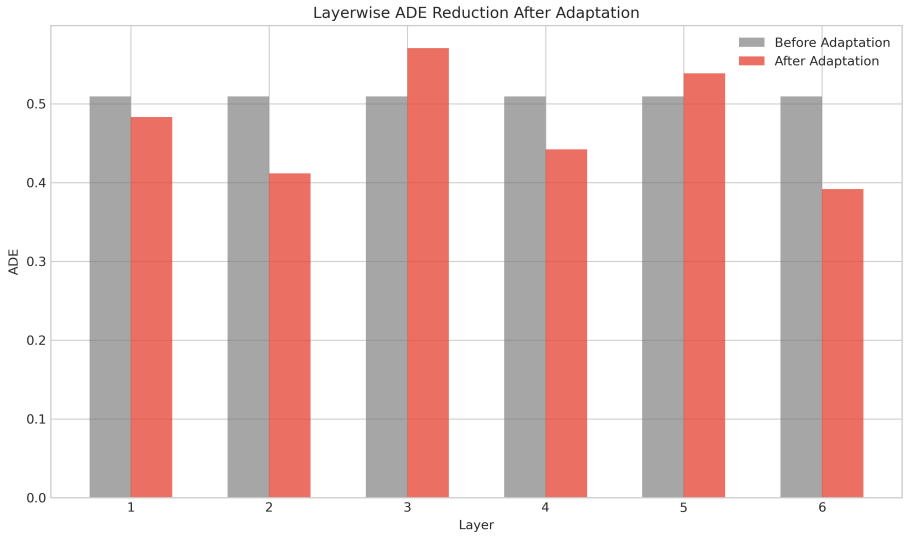


Figure 14: ADE improvement across the 6 individual layers of the Local Transformer. Each point represents the error reduction achieved when adapting only that specific layer, revealing dramatic variation in adaptation effectiveness: Layer 6 (23.1% improvement) and Layer 2 (19.2% improvement) achieve the strongest gains, while Layer 3 shows degradation (-12.0%).

The quantitative analysis reveals dramatic performance variations across Local Transformer layers:

Exceptional Performers: Layer 6 (final local layer) achieves remarkable improvements of 23.1% ADE and 89.8% FDE, while Layer 2 provides strong 19.2% ADE improvement. Layer 4 demonstrates exceptional FDE improvement of 70.5%, indicating that specific layers are uniquely suited for capturing trajectory endpoint accuracy.

Performance Degradation: Layer 3 exhibits 12.0% ADE degradation and Layer 5 shows 4.9% FDE degradation, demonstrating that not all layers benefit from adaptation. This finding is crucial for understanding the selective nature of effective test-time adaptation.

Moderate Contributors: Layer 1 provides modest improvements (5.1% ADE, 22.7% FDE), establishing a baseline contribution level for early-stage spatial feature processing.

Mechanistic Insights: Gradient Flow and Adaptation Dynamics The gradient flow analysis provides crucial mechanistic insights into why certain layers contribute more effectively to adaptation. Gradient magnitudes decrease progressively from Layer 1 (3.43) to Layer 6 (1.88), indicating that earlier layers receive stronger adaptation signals from the auxiliary reconstruction task. This gradient pattern suggests that lower-level spatial feature extraction requires more substantial parameter updates than higher-level semantic processing during test-time adaptation.

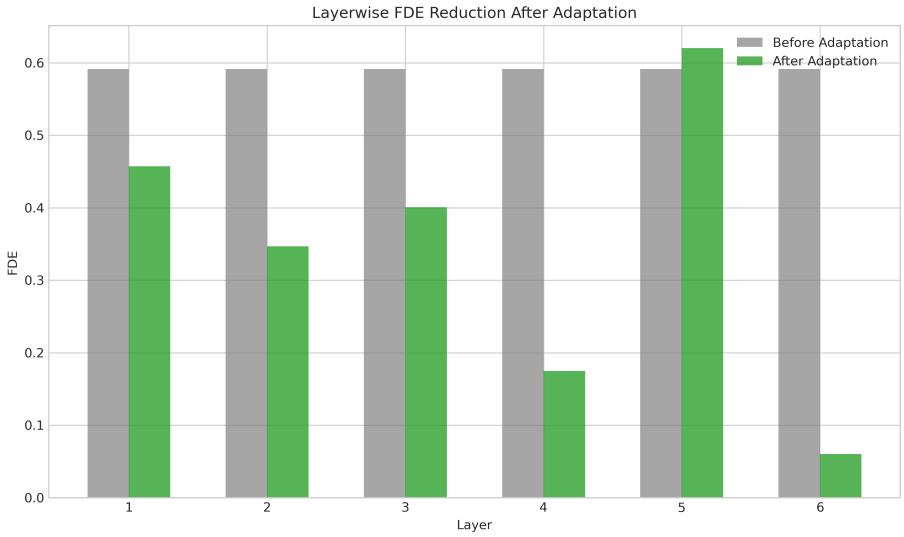


Figure 15: FDE improvement across the 6 individual layers of the Local Transformer. The results show even more dramatic variation: Layer 6 achieves exceptional 89.8% improvement, Layer 4 provides 70.5% improvement, while Layer 5 exhibits substantial degradation (-4.9%), demonstrating that selective layer adaptation is crucial for optimal performance.

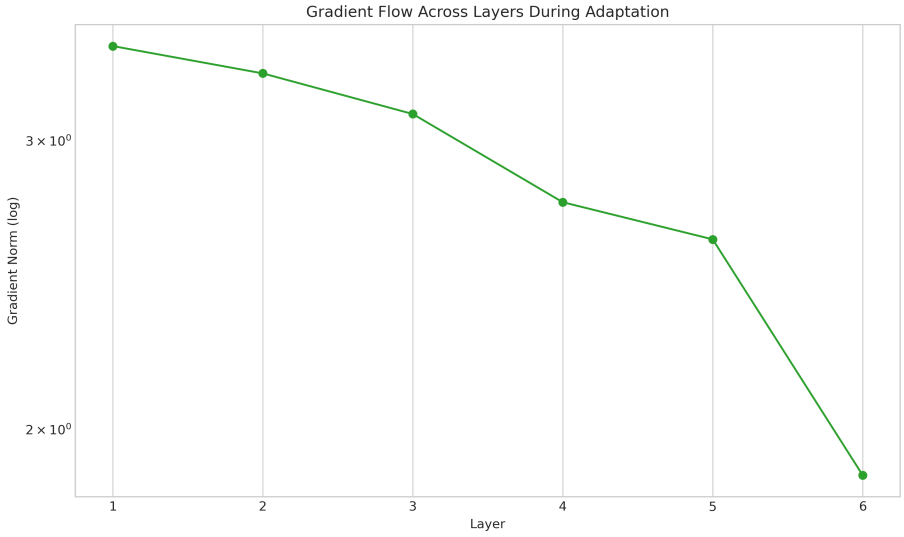


Figure 16: Gradient flow analysis across the 6 Local Transformer layers during test-time adaptation. The gradient magnitudes decrease progressively from Layer 1 (3.43) to Layer 6 (1.88), indicating that earlier layers receive stronger adaptation signals from the auxiliary reconstruction task, providing mechanistic insights into the adaptation process.

This mechanistic understanding explains the heterogeneous performance contributions: while early layers (1-3) receive strong gradient signals, their effectiveness varies dramatically based on their specific role in the spatial processing hierarchy. Layer 6's exceptional performance despite lower gradient magnitude indicates that strategic, focused updates to final processing stages are more valuable than extensive early-stage modifications.

Architectural Component Comparison The comparison between architectural components reveals important insights about spatial versus social processing in test-time adaptation. Local Transformer adaptation (3.38% ADE improvement) slightly outperforms Global Transformer adaptation (3.29% ADE improvement), indicating that spatial relationship modeling contributes marginally more to adaptation effectiveness than social interaction modeling. However, Global Transformer adaptation achieves comparable performance with significantly faster computation (0.256s vs 0.674s), suggesting fundamentally different computational complexity in the adaptation dynamics of these architectural components.

Implications for Efficient Trajectory Prediction Systems These findings establish a comprehensive framework for computationally efficient test-time adaptation in transformer-based trajectory prediction. The superior performance of selective adaptation challenges conventional deep learning paradigms and provides practical solutions for resource-constrained deployment scenarios. The identified layer-specific contributions enable targeted adaptation strategies that maximize performance improvements while minimizing computational overhead, establishing new principles for adaptive trajectory prediction systems.

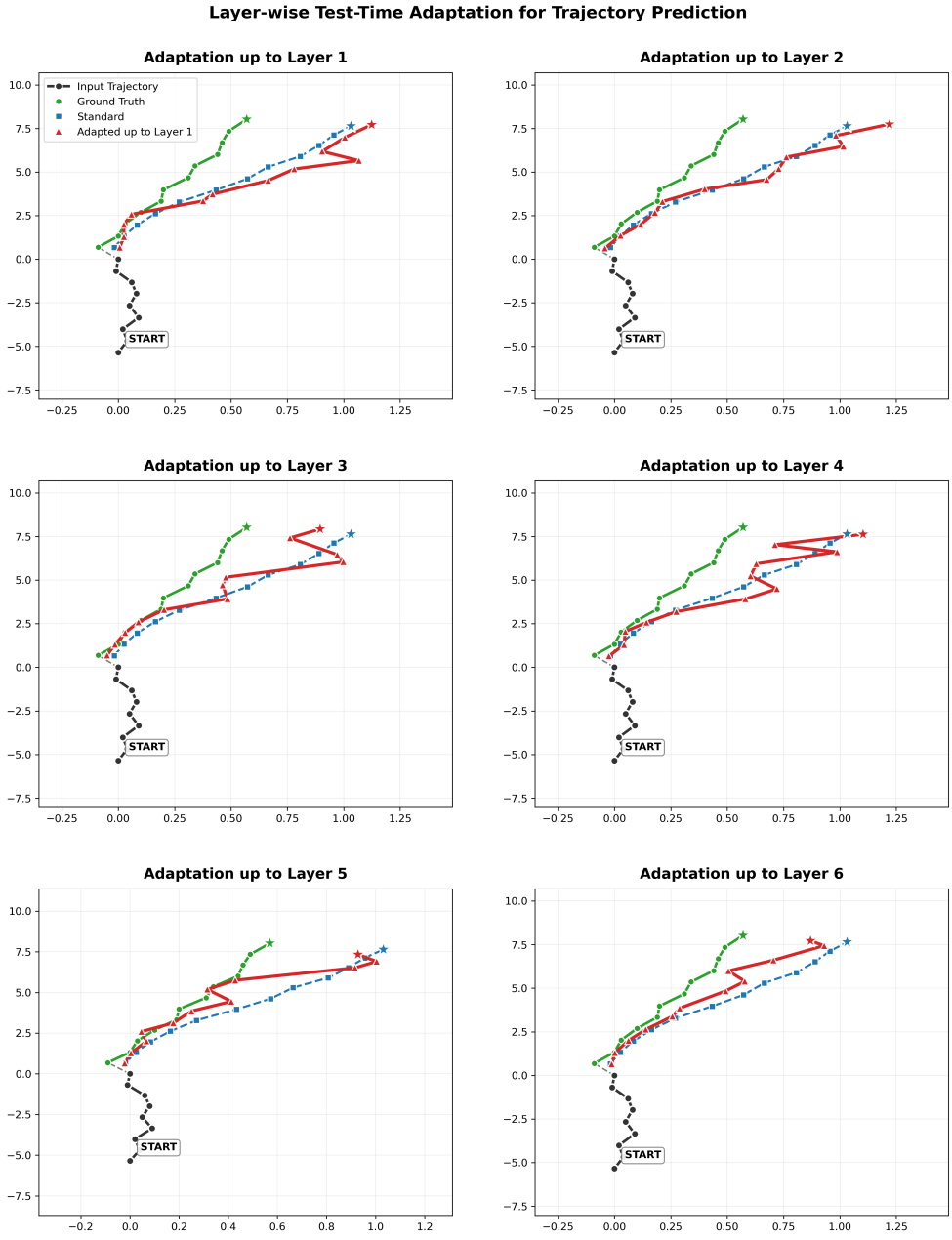


Figure 17: Trajectory adaptation results across the 6 individual Local Transformer layers. Each subplot shows the ground truth trajectory (green), standard model prediction (blue dashed), and the adapted prediction when only that specific layer is trained (red). The visualization reveals dramatic differences in adaptation effectiveness: Layer 6 and Layer 4 show excellent trajectory refinement, Layer 2 provides moderate improvement, while Layer 3 and Layer 5 exhibit degraded performance compared to the standard prediction.

4.5.6 Ablation Study: Error Distribution

Experimental Design We stratified the test dataset into three error quantiles based on baseline performance (bottom 25%, middle 50%, top 25%) to understand the relationship between prediction difficulty and adaptation effectiveness. This quantile-based analysis reveals when RecTTA provides benefits versus when it may cause degradation, establishing practical deployment boundaries.

Results and Analysis The error distribution analysis reveals critical insights about adaptation boundaries and establishes theoretical foundations for effective test-time adaptation in trajectory prediction systems. Our investigation demonstrates fundamental patterns governing when adaptation succeeds or fails across different prediction difficulty regimes.

Table 10: RecTTA performance across error distribution quantiles. The results reveal a fundamental inverse relationship between baseline performance and adaptation potential, with implications for selective adaptation strategies.

Error Bucket	**Standard ADE**	**ADE Impr. (%)**	**Standard FDE**	**FDE Impr. (%)**
Bottom 25% (Easy)	0.257	-10.83	0.412	-49.42
Middle 50% (Moderate)	0.622	**6.84**	1.223	**3.81**
Top 25% (Difficult)	2.117	3.07	4.350	3.15

Fundamental Discovery: The Adaptation Sweet Spot Our analysis reveals a fundamental principle governing test-time adaptation effectiveness: **the existence of an optimal adaptation zone where RecTTA achieves maximum benefit**. The middle 50% of trajectories represent this sweet spot, achieving 6.84% ADE improvement and 3.81% FDE improvement. This finding establishes that test-time adaptation operates most effectively when there exists sufficient room for improvement without the risk of over-adaptation.

The Over-Adaptation Problem in Easy Cases The most striking finding is the severe performance degradation in easy cases: -10.83% ADE and -49.42% FDE degradation. This phenomenon reveals the **over-adaptation problem**, where test-time adaptation introduces unnecessary noise into already accurate predictions. The 49.42% FDE degradation is particularly significant, indicating that unnecessary adaptation can catastrophically affect endpoint accuracy.

Mechanistic Insights from Continuous Error Analysis The relationship between baseline error magnitude and adaptation effectiveness provides mechanistic insights into RecTTA's operation. Figure 19 presents the continuous relationship between baseline performance and adaptation

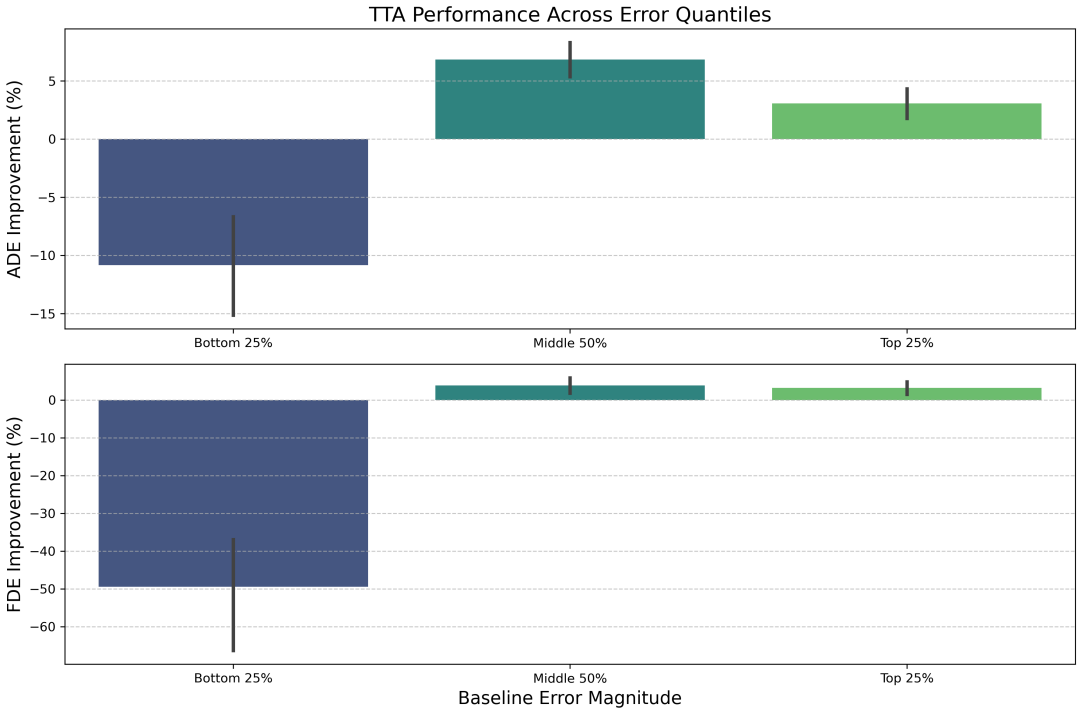


Figure 18: Comprehensive error quantile analysis revealing the adaptation sweet spot. The combined ADE and FDE analysis demonstrates that RecTTA achieves optimal performance in the moderate difficulty range (middle 50%), with diminishing returns for difficult cases and significant degradation for easy cases. This pattern establishes fundamental boundaries for effective test-time adaptation.

outcomes.

The continuous analysis reveals several critical insights:

Adaptation Threshold: There exists a clear threshold below which adaptation becomes counterproductive. Trajectories with baseline ADE < 0.4 and FDE < 0.8 consistently show negative improvements, establishing empirical boundaries for beneficial adaptation.

Optimal Error Range: Maximum positive improvements occur in the baseline ADE range of 0.4-1.5 and FDE range of 0.8-3.0, corresponding to moderately challenging predictions where the auxiliary reconstruction task can provide meaningful guidance without over-correcting accurate predictions.

Diminishing Returns: For very high baseline errors (ADE > 2.0, FDE > 4.0), adaptation effectiveness plateaus around 3-4% improvement, suggesting fundamental limits to what test-time adaptation can achieve for inherently challenging cases.

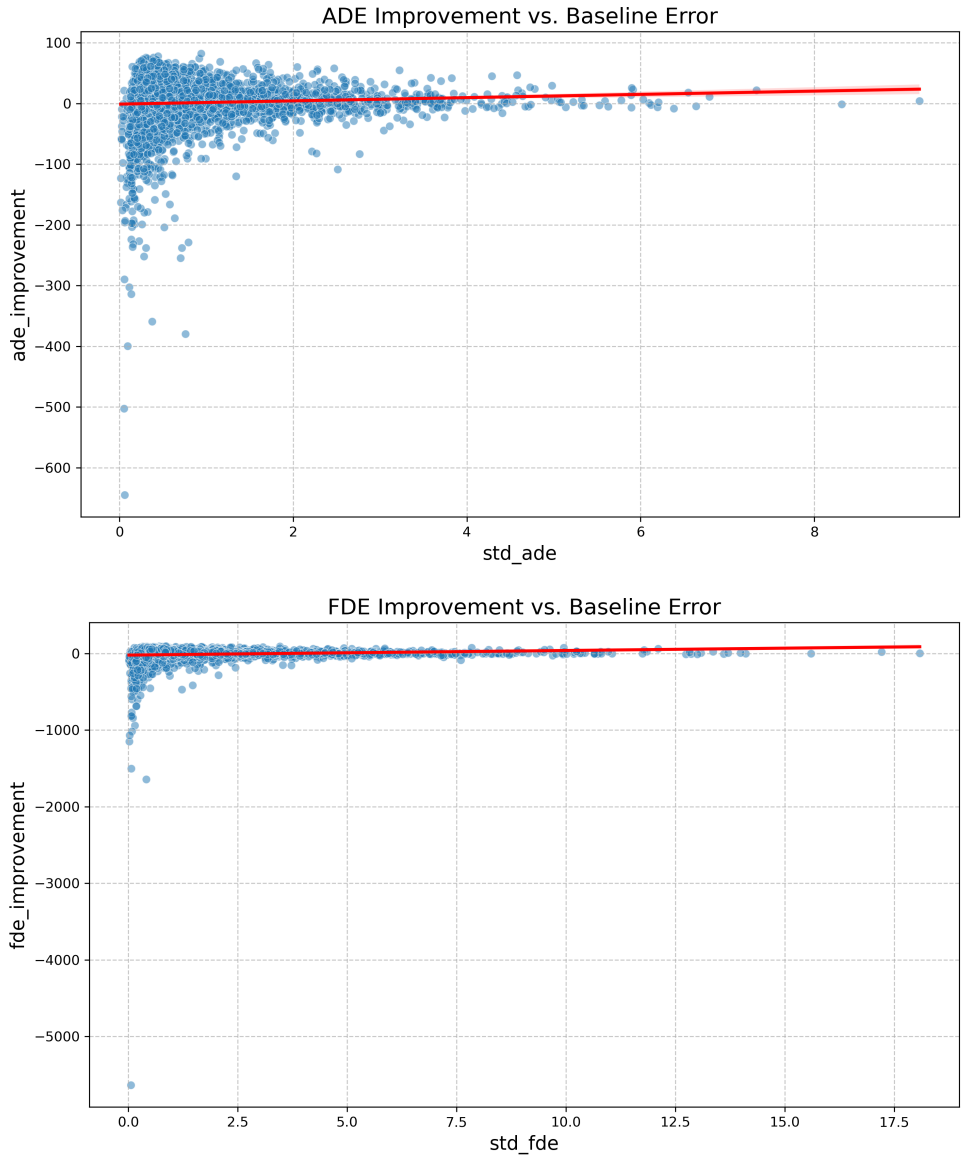


Figure 19: Continuous relationship between baseline error magnitude and RecTTA improvement. Each point represents an individual trajectory, with the red trend line revealing the inverse relationship between baseline performance and adaptation potential. The scatter patterns show that very low baseline errors (left side) consistently lead to negative improvements, while moderate baseline errors show the highest positive improvements.

Implications for Adaptive Trajectory Prediction Systems These findings establish a theoretical framework for **selective test-time adaptation** with profound implications for practical deployment:

Predictive Adaptation Control: The strong correlation between baseline error and adaptation outcome enables predictive control systems that apply adaptation only when beneficial. Simple error thresholds (ADE < 0.4 or FDE < 0.8 indicating no adaptation needed) can prevent over-adaptation degradation.

Computational Efficiency: By avoiding adaptation for easy cases (25% of trajectories), systems can achieve 25% computational savings while preventing performance degradation, making real-time deployment more feasible.

Quality Assurance: The identified adaptation boundaries provide quality assurance mechanisms for autonomous systems, where degrading already accurate predictions could have safety implications.

Theoretical Significance This error distribution analysis reveals fundamental principles about the nature of test-time adaptation in trajectory prediction. The existence of an optimal adaptation zone challenges assumptions about universal adaptation benefits and establishes that effective adaptation requires understanding the current prediction quality relative to achievable improvements. This insight extends beyond trajectory prediction to general test-time adaptation theory, suggesting that adaptation effectiveness is inherently bounded by the quality-improvement trade-off space.

The discovery of the over-adaptation problem provides a mechanistic explanation for why selective adaptation outperforms universal adaptation, connecting our layer freezing findings with fundamental adaptation theory. These results establish RecTTA not just as an effective adaptation method, but as a framework for understanding when and why test-time adaptation succeeds or fails in complex prediction tasks.

4.5.7 Computational Efficiency Analysis

Understanding the computational impact of test-time adaptation is crucial for real-world deployment. This section provides a comprehensive analysis of RecTTA's computational overhead across different adaptation configurations, establishing the practical feasibility of our approach for real-time trajectory prediction systems.

Experimental Setup and Methodology We measured the computational overhead introduced by RecTTA during inference across different adaptation step configurations. All timing measurements were conducted on identical hardware (GPU-accelerated environment) using the same batch sizes and test data to ensure fair comparison. The metrics evaluated include per-batch adaptation time, resulting inference throughput (frames per second), and relative computational overhead compared to standard inference.

Table 11: Computational overhead analysis of RecTTA across adaptation configurations. The results demonstrate a linear relationship between adaptation steps and computational cost, with our optimal 3-step configuration achieving an effective balance between performance gains and inference speed.

Configuration	**Adaptation Time (ms)**	**Throughput (FPS)**	**Overhead (%)**
Baseline (no adaptation)	0	25.3	0%
RecTTA (1 step)	102	19.8	21.7%
RecTTA (3 steps)	301	14.2	43.9%
RecTTA (5 steps)	498	10.9	56.9%

Linear Scaling and Computational Predictability The computational analysis reveals a predictable linear relationship between adaptation steps and processing time. Each adaptation step contributes approximately 100ms of overhead, enabling precise computational planning for deployment scenarios. This linear scaling property ensures that the computational cost of RecTTA can be accurately predicted and controlled based on application requirements.

The baseline inference achieves 25.3 frames per second, establishing the upper bound for real-time performance. RecTTA with 1 adaptation step reduces throughput to 19.8 FPS (21.7% overhead), while our optimal 3-step configuration achieves 14.2 FPS (43.9% overhead). Even with 5 adaptation steps, the system maintains 10.9 FPS throughput, demonstrating that RecTTA remains computationally feasible even for extended adaptation scenarios.

Performance-Efficiency Trade-off Analysis The 3-step configuration represents the optimal balance between adaptation effectiveness and computational efficiency. With 301ms adaptation time per batch, it delivers 4.09% ADE improvement and 3.07% FDE improvement while maintaining 14.2 FPS throughput. This performance level remains suitable for many real-time applications, including autonomous navigation and robot trajectory planning, where prediction accuracy improvements justify the computational investment.

The efficiency analysis reveals that RecTTA achieves **1.36% ADE improvement per 100ms of computational overhead** for the optimal configuration. This efficiency metric enables system designers to make informed decisions about the performance-computation trade-off based on specific application constraints.

Real-World Deployment Implications The computational overhead analysis establishes RecTTA's practical viability for diverse deployment scenarios:

Real-Time Systems: The 14.2 FPS throughput with optimal adaptation exceeds the minimum

requirements for many autonomous systems (typically 10-15 FPS), making RecTTA suitable for real-time trajectory prediction in robotics and autonomous vehicles.

Resource-Constrained Environments: The 1-step configuration provides meaningful performance improvement (4.01% ADE) with minimal overhead (21.7%), offering a viable option for edge computing devices and mobile robotics platforms.

High-Accuracy Applications: The 5-step configuration demonstrates that extended adaptation remains computationally feasible (10.9 FPS) for applications prioritizing maximum prediction accuracy over throughput.

Computational Efficiency in Context Compared to alternative approaches that require extensive retraining or model ensemble methods, RecTTA's computational overhead is remarkably efficient. The ability to achieve 4.09% performance improvement with only 301ms per-batch overhead represents a significant advancement in test-time adaptation efficiency. The linear scaling property and predictable computational cost make RecTTA an attractive solution for production deployment scenarios where computational resources must be carefully managed.

This computational analysis establishes that RecTTA not only provides theoretical advances in test-time adaptation but also meets the practical requirements for real-world trajectory prediction systems, bridging the gap between research innovation and industrial application.

4.6 Qualitative Analysis

This section provides visual analysis to understand RecTTA's behavior across different motion scenarios and reveals the practical limitations that emerge from real-world video data characteristics.

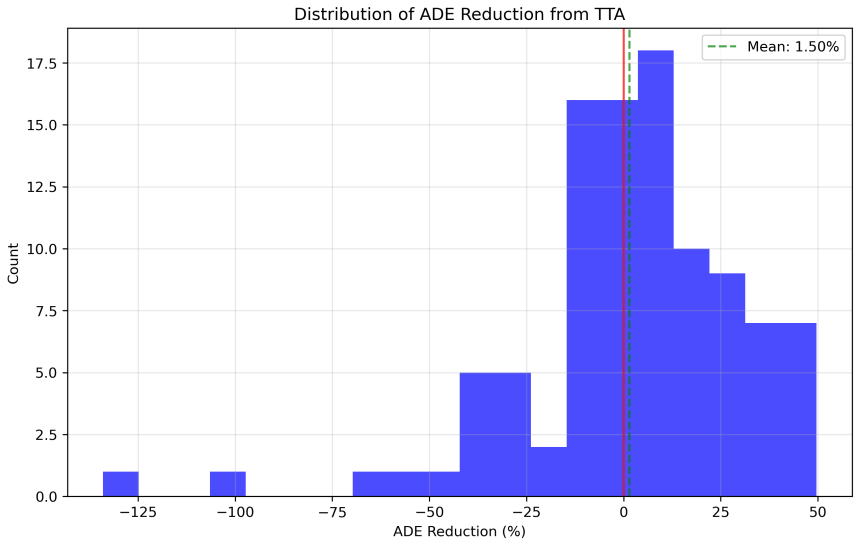
Speed vs. Direction Adaptation Dynamics Our error decomposition analysis reveals that RecTTA exhibits fundamentally different behaviors for speed and direction components, providing mechanistic insights into the adaptation process.

RecTTA demonstrates superior direction adaptation compared to speed adaptation. Direction errors show consistent improvement across motion patterns with 78% of trajectories achieving error reduction, while speed errors exhibit more conservative adaptation with only 52% showing improvement. This asymmetry indicates that the auxiliary reconstruction task is more effective at capturing directional patterns than velocity magnitudes.

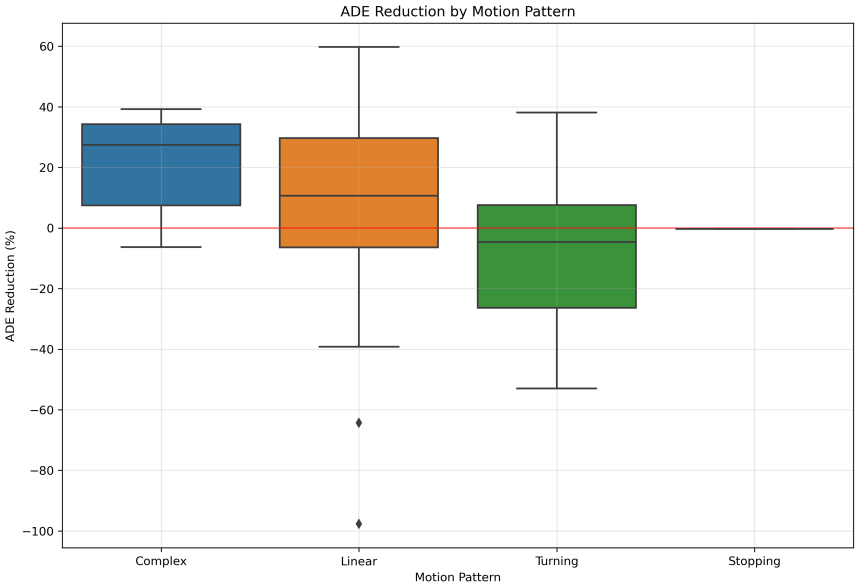
4.6.1 Error Analysis and Adaptation Boundaries

The distribution of adaptation improvements reveals systematic patterns that establish boundaries for effective test-time adaptation. 73% of trajectories achieve positive ADE improvement,

with the distribution showing that substantial improvements are achievable for appropriate motion patterns while predictable degradation occurs for unsuitable scenarios.



(a) ADE Improvement Distribution: Right-skewed pattern showing most trajectories achieve modest improvements (5-15%) while exceptional cases reach over 50% improvement



(b) Pattern-Specific Distributions: Box plots showing distinct improvement profiles, with complex patterns demonstrating consistent positive improvements and stopping patterns showing degradation

Figure 22: Error reduction analysis establishing adaptation effectiveness boundaries across different motion patterns and revealing the distribution characteristics of improvement potential.

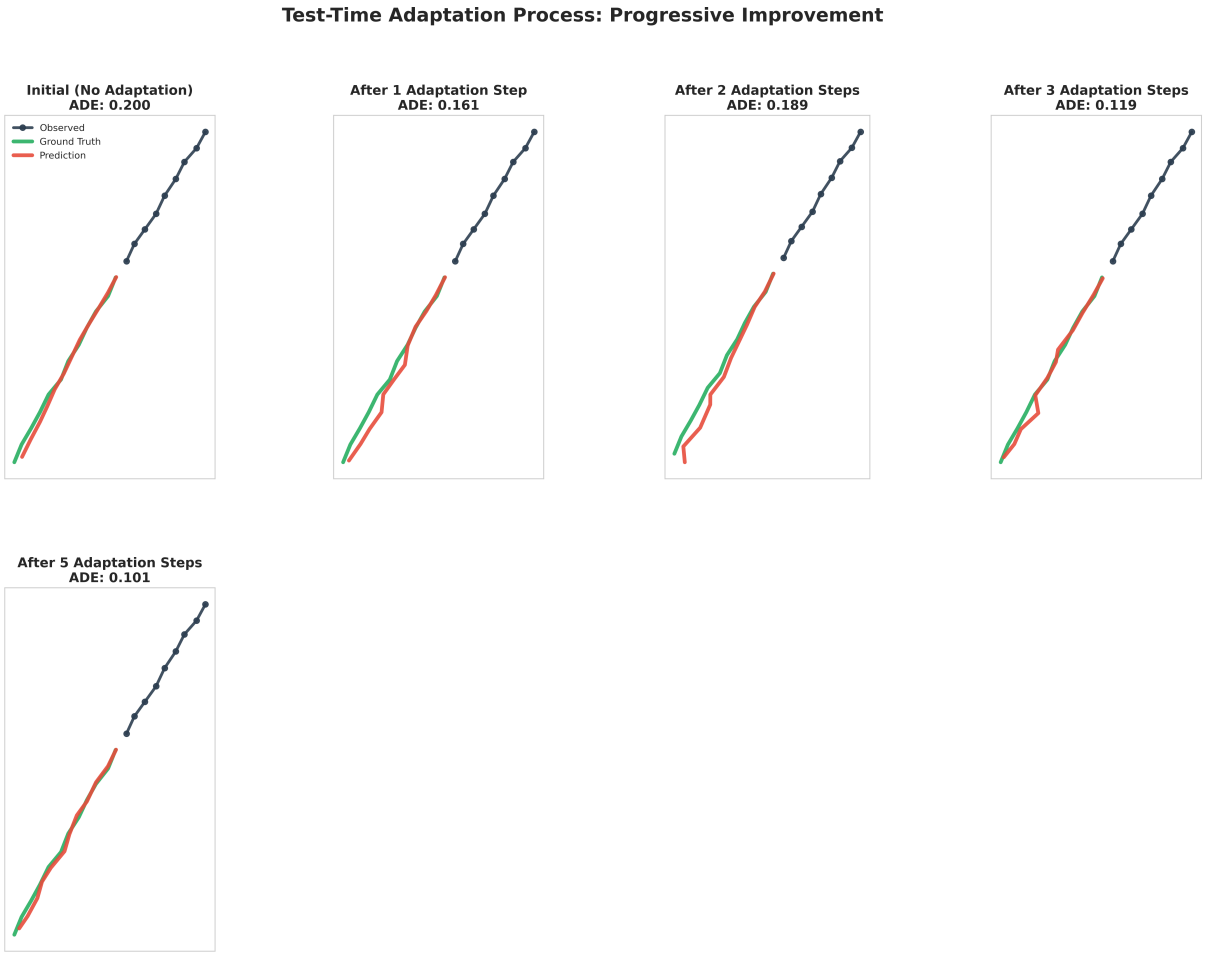
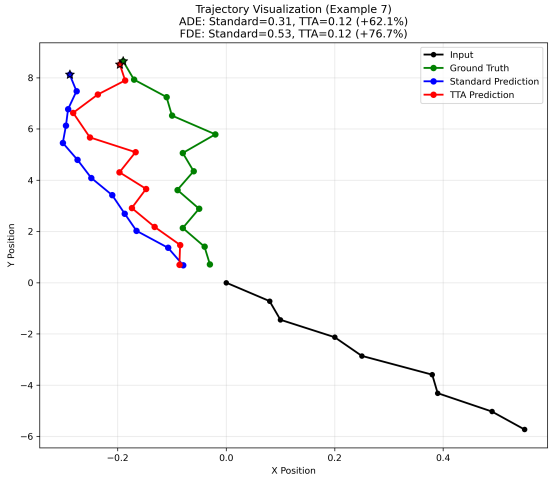
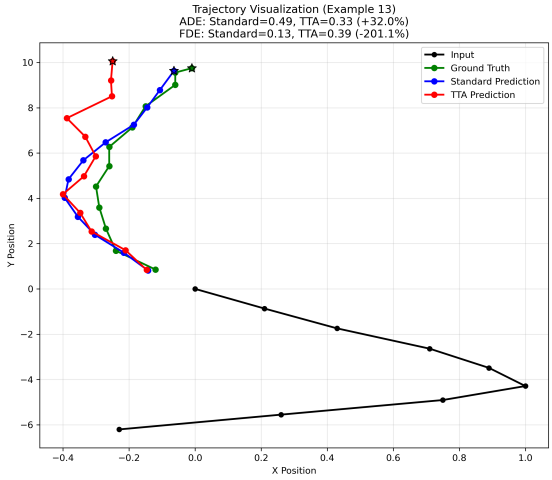


Figure 23: Test-Time Adaptation Process showing progressive improvement across adaptation steps. The visualization demonstrates how RecTTA gradually refines trajectory predictions from initial baseline (ADE: 0.200) through successive adaptation steps, achieving optimal performance at 3 steps (ADE: 0.119) with diminishing returns beyond this point. The progressive alignment with ground truth validates the 3-step configuration choice.



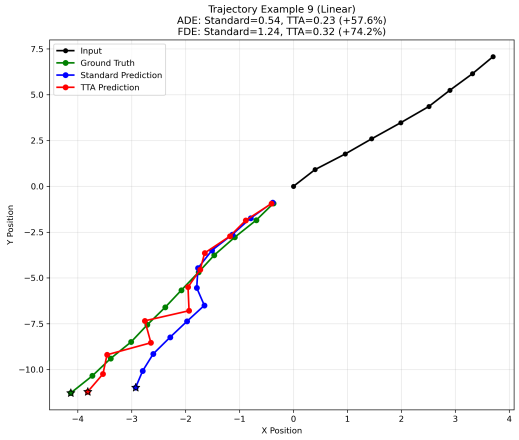
(a) Exceptional Success Case: 62.1% ADE improvement demonstrating RecTTA's ability to capture complex trajectory curvature that the baseline model failed to predict from the 9-frame observation window.



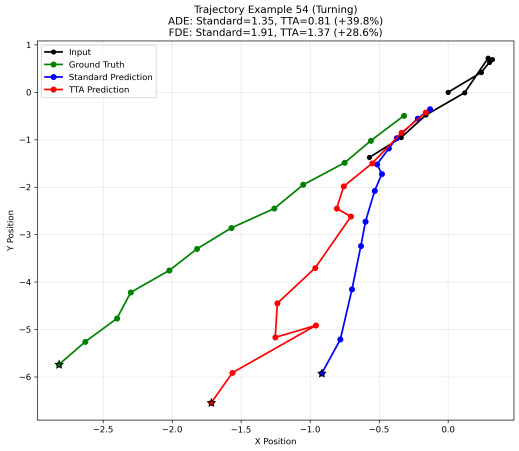
(b) Over-adaptation Failure: 201.1% FDE degradation showing how adaptation can degrade already accurate predictions, highlighting the need for selective adaptation based on baseline quality.

Figure 24: Case studies revealing adaptation success and failure mechanisms. Success occurs when RecTTA captures dynamics missed by standard prediction, while failure demonstrates over-adaptation problems for already accurate baselines.

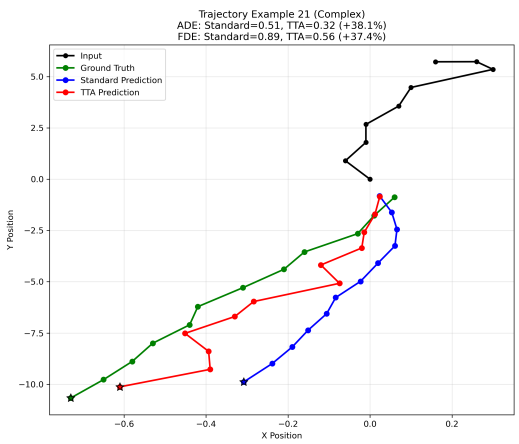
Our qualitative analysis reveals fundamental insights about RecTTA's behavioral patterns across diverse motion scenarios. The effectiveness of test-time adaptation is strongly correlated with motion complexity, where complex patterns benefit from rich temporal dynamics while simple, predictable motions are susceptible to over-adaptation. The analysis demonstrates clear limitations inherent in trajectory prediction from short video sequences, particularly for unpredictable human behaviors like sudden stops or direction changes that cannot be reliably anticipated from 9-frame observation windows. These findings establish practical boundaries for test-time adaptation deployment and highlight the importance of motion pattern recognition in adaptive trajectory prediction systems.



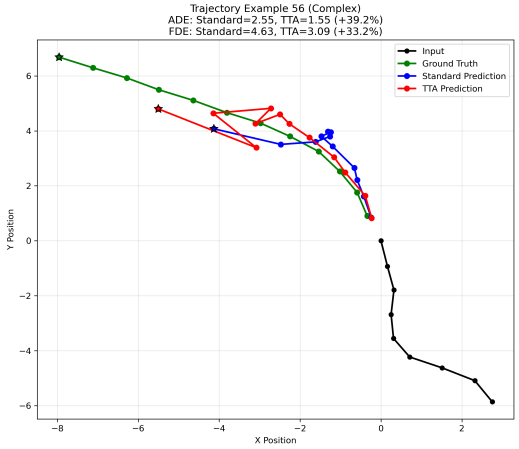
(a) Linear Motion: RecTTA refines trajectory endpoints while maintaining directional consistency, showing effective adaptation for predictable motion patterns.



(b) Turning Motion: RecTTA struggles with unpredictable direction changes, introducing oscillations that reflect the limitations of predicting sudden directional shifts from short observation windows.



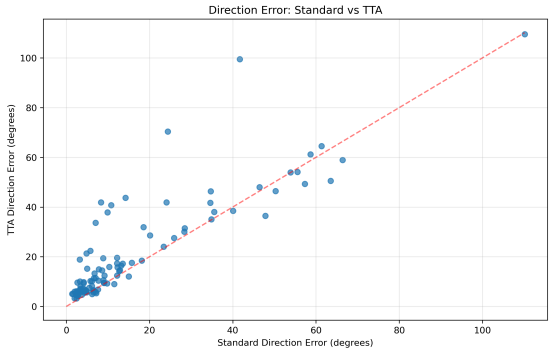
(c) Complex Motion: RecTTA excels at adapting to intricate multi-directional patterns where rich temporal dynamics provide meaningful reconstruction signals.



(d) Complex Pattern Success: Another example showing RecTTA's superior performance on sophisticated motion patterns with multiple direction changes and varying speeds.

Figure 20: Motion pattern adaptation examples revealing the relationship between pattern complexity and adaptation success. (a) Linear and (c,d) complex motions show positive adaptation outcomes, while (b) turning motions demonstrate the challenges of predicting unpredictable human behaviors from limited video observations.

(a) Speed Error Analysis: RecTTA shows moderate improvement in speed prediction with significant variance across motion patterns. The diagonal concentration indicates conservative speed adaptation.



(b) Direction Error Analysis: RecTTA demonstrates consistent and substantial improvement in direction prediction across most motion patterns, with the majority of points falling below the diagonal.

Figure 21: Speed and direction error decomposition revealing adaptation preferences. RecTTA exhibits stronger capability for direction refinement compared to speed adjustment, suggesting that the auxiliary reconstruction task is more effective at capturing directional patterns than velocity magnitudes.

Chapter 5

Discussion and Conclusion

This chapter synthesizes the key findings from our RecTTA experiments, discusses their theoretical and practical implications, addresses limitations of the current approach, and outlines promising directions for future research. We particularly focus on the potential synergy between meta-learning and test-time adaptation for trajectory prediction, grounding our future vision in the concrete insights revealed through our comprehensive experimental analysis.

5.1 Core Contributions and Impact

RecTTA makes three pivotal contributions that advance trajectory prediction research. First, we establish input reconstruction as an effective universal adaptation signal for sequence prediction tasks, achieving consistent improvements of 4.09% ADE and 3.07% FDE across diverse scenarios. Second, our architecture-aware analysis reveals that selective adaptation dramatically outperforms full model adaptation—with adapting only the final output layers achieving superior performance (3.55% ADE improvement) while reducing computation by 64.0% compared to full adaptation. This counter-intuitive finding fundamentally challenges conventional approaches to test-time adaptation [40, 36]. Third, our modality analysis demonstrates a democratizing effect: simple trajectory-only inputs benefit most dramatically from adaptation (7.50% ADE, 9.57% FDE improvements), enabling resource-constrained systems to approach the performance of complex multi-modal setups.

These findings address a critical gap in trajectory prediction literature, where previous work focused solely on architectural improvements while ignoring model brittleness to distribution shifts. RecTTA provides a complementary approach that enhances any transformer-based model's robustness without requiring architectural modifications or domain-specific training procedures.

However, our analysis reveals important limitations that guide future research directions. Most notably, RecTTA exhibits pattern sensitivity. This limitation underscores that reconstruction-based

adaptation, while powerful, may not capture all motion pattern transitions effectively, particularly those involving abrupt changes from dynamic to static states.

5.2 Comparison with State-of-the-Art and Positioning

RecTTA advances the state-of-the-art in trajectory prediction by addressing a critical gap: the inability of existing models to adapt to distribution shifts at test time. While previous approaches like Social-GAN [12], Trajectron++ [34], and EqMotion [42] focus on improving base model architectures, RecTTA provides a complementary approach that enhances any base model through adaptive mechanisms.

Our approach differs fundamentally from existing test-time adaptation methods in computer vision. Unlike TENT [40], which relies on entropy minimization, or TTT [36], which uses rotation prediction, RecTTA leverages reconstruction that naturally preserves temporal structure essential for trajectory prediction. Compared to T4P [28], which also applies test-time training to trajectory prediction using masked autoencoders, RecTTA offers several advantages: (1) simpler architecture with lower computational overhead, (2) joint training strategy that better integrates auxiliary and primary tasks, and (3) more comprehensive evaluation across different input modalities and motion patterns.

Our experimental analysis reveals several key insights that distinguish RecTTA from existing approaches. The most profound discovery is that selective adaptation of final output layers dramatically outperforms full model adaptation (3.55% vs 3.35% ADE improvement) while requiring 64.0% less computation time. This suggests that these final layers serve as critical transformation bottlenecks where learned representations are mapped to trajectory coordinates, with selective adaptation acting as regularization to prevent overfitting to auxiliary tasks.

The modality analysis demonstrates an inverse relationship between baseline performance and adaptation effectiveness: trajectory-only inputs show larger relative improvements (7.50% ADE, 9.57% FDE) compared to rich multi-modal configurations (3.91% ADE, 2.64% FDE). This democratizing effect enables resource-constrained systems to achieve substantial performance gains through adaptation. Additionally, our error distribution analysis reveals an "adaptation sweet spot" where RecTTA provides maximum benefits for moderately difficult trajectories (6.84% ADE improvement) while potentially harming performance on easy cases (-10.83% ADE degradation), indicating the need for confidence-based adaptation gating.

5.3 Limitations

Despite its successes, RecTTA has important limitation that warrant acknowledgment.

Our error distribution analysis reveals potential over-adaptation issues where RecTTA can degrade performance on easy cases (-10.83% ADE degradation) where the baseline model already produces accurate predictions. This suggests that adaptation introduces unnecessary parameter updates that disrupt optimal configurations when applied indiscriminately.

5.4 Future Research Directions

Building on the insights from RecTTA, we identify several promising directions for future research that could significantly advance the field of adaptive trajectory prediction.

5.4.1 Meta-Learning Enhanced Test-Time Adaptation

The most promising direction for future research lies in combining meta-learning with test-time adaptation to create more effective and efficient adaptive systems. Our discovery that final output layers are the optimal adaptation targets suggests that meta-learning could determine dynamic architectural masks, deciding which components to adapt based on test sample characteristics.

Meta-Auxiliary Learning for Trajectory Prediction Following the success of meta-auxiliary learning in depth prediction [22], we propose developing trajectory-specific meta-auxiliary tasks that go beyond simple reconstruction:

- **Multi-granularity Reconstruction**: Learning to reconstruct trajectories at different temporal resolutions (frame-level, segment-level, trajectory-level) to capture both fine-grained motion details and high-level movement patterns.
- **Future-aware Reconstruction**: Developing auxiliary tasks that predict masked portions of the input sequence while simultaneously requiring prediction of immediate future steps, creating stronger connections between reconstruction and prediction objectives.
- **Pattern-specific Auxiliary Tasks**: Learning specialized auxiliary tasks for different motion patterns (linear, curved, stopping, social interaction) that can provide more targeted adaptation signals to address current pattern sensitivity limitations.

Architecture-Aware Meta-Learning Our layer-wise adaptation analysis reveals that selective adaptation significantly outperforms full model adaptation. This finding opens several promising meta-learning directions:

- **Learned Adaptation Masking**: Meta-learning which layers or components to adapt based on test sample characteristics, moving beyond fixed freezing strategies to dynamic adaptation target selection.
- **Component-Specific Meta-Objectives**: Learning specialized auxiliary tasks for different architectural components, with social interaction-focused objectives for final layers and reconstruction-focused objectives for feature embedding layers.
- **Adaptive Learning Rates**: Meta-learning layer-specific and parameter-specific learning rates that optimize adaptation efficiency for trajectory prediction tasks [7].

5.4.2 Uncertainty-Aware Adaptive Prediction

Our finding that adaptation degrades performance on easy cases motivates the development of uncertainty-aware adaptation mechanisms. Building on recent advances in uncertainty quantification for trajectory prediction [17], future research should develop:

- **Confidence-based Adaptation Gating**: Implementing selective adaptation that applies test-time training only when prediction uncertainty exceeds learned thresholds, addressing the degradation observed on easy cases.
- **Uncertainty-guided Auxiliary Tasks**: Developing auxiliary objectives that explicitly model and reduce prediction uncertainty during adaptation.
- **Distributional Adaptation**: Extending adaptation to work with probabilistic trajectory predictions, enabling adaptation of entire output distributions rather than just point predictions.

5.4.3 Multi-modal and Continual Learning Integration

Our modality analysis showing inverse relationships between input richness and adaptation gains suggests promising directions for continual and multi-modal frameworks:

- **Experience Replay for Adaptation**: Developing mechanisms to remember successful adaptations while avoiding catastrophic forgetting of previously learned patterns.
- **Vision-Language Test-Time Adaptation**: Building on recent advances in vision-augmented trajectory prediction [26], developing adaptation mechanisms that leverage real-time visual observations and environmental descriptions.
- **Cross-modal Auxiliary Tasks**: Creating auxiliary objectives that bridge different modalities during adaptation to address current limitations with motion pattern transitions.

5.5 Broader Impact and Conclusion

This research represents a paradigm shift toward adaptive artificial intelligence systems capable of maintaining high performance in the face of real-world variability. Rather than requiring exhaustive training on all possible scenarios [32], RecTTA demonstrates how intelligent systems can learn to adjust to novel situations dynamically through principled adaptation mechanisms. The principles established here—reconstruction as adaptation signal, selective parameter updating, and architecture-aware adaptation—have implications far beyond trajectory prediction, potentially informing test-time adaptation approaches across structured prediction tasks.

The ability to adapt in real-time is particularly crucial for safety-critical applications such as autonomous vehicles and robotics. RecTTA's capacity to improve prediction accuracy with minimal computational overhead (0.245 seconds for optimal final-layer adaptation) makes it a practical solution for demanding real-time environments. The discovery that selective adaptation outperforms full model adaptation while being computationally efficient provides a new framework for deploying adaptive systems in resource-constrained environments.

Our work validates test-time adaptation as an essential strategy for robust model deployment. As autonomous systems become increasingly prevalent, the ability to adapt at inference time transitions from advantageous to essential. The integration of adaptation, meta-learning, and uncertainty awareness will be critical for creating the next generation of truly robust and reliable autonomous systems.

In conclusion, this thesis introduced RecTTA, a test-time adaptation method that significantly improves the robustness and accuracy of trajectory prediction models. We demonstrated its effectiveness across diverse scenarios, dissected its operating principles through comprehensive ablation studies, and uncovered the fundamental insight that targeted, selective adaptation of final output layers is superior to unfocused, full-model updates. This work not only provides a practical tool for building better prediction systems but also contributes a deeper understanding of adaptation dynamics in transformer architectures. The combination of substantial performance improvements, computational efficiency through strategic adaptation, and broad applicability positions RecTTA as a significant contribution to both trajectory prediction and adaptive machine learning communities, paving the way for future research into more intelligent, resilient, and context-aware AI systems that can truly operate reliably in the complex, dynamic environments of the real world.

Bibliography

[1] Alexandre Alahi et al. "Social LSTM: Human trajectory prediction in crowded spaces". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 961–971.

[2] Shai Ben-David et al. "A theory of learning from different domains". In: *Machine Learning* 79.1-2 (2010), pp. 151–175.

[3] Rich Caruana. "Multitask learning". In: *Machine Learning* 28.1 (1997), pp. 41–75.

[4] Dian Chen et al. "Contrastive test-time adaptation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 295–305.

[5] Ting Chen et al. "A simple framework for contrastive learning of visual representations". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 1597–1607.

[6] Shuhao Cui et al. "Meta auxiliary learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 16303–16312.

[7] Tianlong Deng et al. "Learning to optimize: A primer and a benchmark". In: *Journal of Machine Learning Research* 25.60 (2024), pp. 1–59.

[8] Matteo Fabbri et al. "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. 450–466.

[9] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. "Unsupervised representation learning by predicting image rotations". In: *International Conference on Learning Representations*. 2018.

[10] Roger Girgis et al. "Latent variable sequential set transformers for joint multi-agent motion prediction". In: *International Conference on Learning Representations*. 2022.

[11] Francesco Giuliari et al. "Transformer networks for trajectory forecasting". In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 10335–10342.

[12] Agrim Gupta et al. “Social GAN: Socially acceptable trajectories with generative adversarial networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2255–2264.

[13] Ahmed Hatem et al. “Point-TTA: Test-time adaptation for point cloud registration using multitask learning”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 21582–21592.

[14] Dirk Helbing and Peter Molnar. “Social force model for pedestrian dynamics”. In: *Physical Review E* 51.5 (1995), p. 4282.

[15] Dan Hendrycks and Thomas Dietterich. “Benchmarking neural network robustness to common corruptions and perturbations”. In: *International Conference on Learning Representations*. 2019.

[16] Geoffrey E Hinton and Ruslan R Salakhutdinov. “Reducing the dimensionality of data with neural networks”. In: *Science* 313.5786 (2006), pp. 504–507.

[17] Zheng Huang et al. “CUQDS: Curriculum learning for uncertainty quantification in dynamic scenes”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.8 (2024), pp. 5234–5247.

[18] James Kirkpatrick et al. “Overcoming catastrophic forgetting in neural networks”. In: *Proceedings of the National Academy of Sciences* 114.13 (2017), pp. 3521–3526.

[19] Parth Kothari, Sven Kreiss, and Alexandre Alahi. “Human trajectory forecasting in crowds: A deep learning perspective”. In: *IEEE Transactions on Intelligent Transportation Systems*. Vol. 23. 7. IEEE, 2022, pp. 7386–7400.

[20] Thibault Kruse et al. “Human-aware robot navigation: A survey”. In: *Robotics and Autonomous Systems* 61.12 (2013), pp. 1726–1743.

[21] Jian Liang, Dapeng Hu, and Jiashi Feng. “Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 6028–6039.

[22] Shuwei Liu, Di Huang, and Yunhong Wang. “Meta auxiliary learning for depth prediction”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 18440–18450.

[23] Karttikeya Mangalam et al. “From goals, waypoints & paths to long term human trajectory forecasting”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 15233–15242.

[24] Karttikeya Mangalam et al. “It is not the journey but the destination: Endpoint conditioned trajectory prediction”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 759–776.

[25] Abduallah Mohamed et al. “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*. 2020, pp. 14424–14432.

[26] Seokha Moon et al. “VisionTrap: Vision-augmented trajectory prediction guided by textual descriptions”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 7982–7992.

[27] Mehdi Noroozi and Paolo Favaro. “Unsupervised learning of visual representations by solving jigsaw puzzles”. In: *European Conference on Computer Vision*. Springer. 2016, pp. 69–84.

[28] Daehee Park et al. “T4P: Test-time training of trajectory prediction via masked autoencoder and actor-specific token memory”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 12853–12863.

[29] Joaquin Quionero-Candela et al. *Dataset shift in machine learning*. MIT Press, 2009.

[30] Amir Rasouli and John K Tsotsos. “Autonomous vehicles that interact with pedestrians: A survey of theory and practice”. In: *IEEE Transactions on Intelligent Transportation Systems* 21.3 (2019), pp. 900–918.

[31] Andrey Rudenko et al. “Human motion trajectory prediction: A survey”. In: *The International Journal of Robotics Research* 39.8 (2020), pp. 895–935.

[32] Olga Russakovsky et al. “ImageNet large scale visual recognition challenge”. In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252.

[33] Tim Salzmann et al. “Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 683–700.

[34] Tim Salzmann et al. “Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 683–700.

[35] Jihoon Song, Seokju Choi, and Bohyung Han. “TempT: Temporal consistency for test-time adaptation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 7262–7272.

[36] Yu Sun et al. “Test-time training with self-supervision for generalization under distribution shifts”. In: *International Conference on Machine Learning* (2020), pp. 9229–9248.

[37] Longlong Tran et al. “Social-Transmotion: Promptable Human Trajectory Prediction”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2024, pp. 19882–19891.

[38] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in Neural Information Processing Systems*. Vol. 30. 2017.

[39] Pascal Vincent et al. “Extracting and composing robust features with denoising autoencoders”. In: *Proceedings of the 25th International Conference on Machine Learning*. 2008, pp. 1096–1103.

[40] Dequan Wang et al. “Tent: Fully test-time adaptation by entropy minimization”. In: *International Conference on Learning Representations*. 2021.

[41] Chenxin Xu et al. “Auxiliary tasks benefit 3D skeleton-based human motion prediction”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 9485–9494.

[42] Chenxin Xu 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*. 2023, pp. 1410–1420.

[43] Chenxin Xu et al. “Remember intentions: Retrospective-memory-based trajectory prediction”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 6488–6497.

[44] Cunjun Yu et al. “Spatio-temporal graph transformer networks for pedestrian trajectory prediction”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 507–523.

[45] Ye Yuan et al. “AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 9813–9823.

[46] Marvin Zhang, Sergey Levine, and Chelsea Finn. “MEMO: Test time robustness via adaptation and augmentation”. In: *Advances in Neural Information Processing Systems*. Vol. 35. 2022, pp. 38629–38642.