https://doi.org/10.2352/EI.2026.38.16.AVM-101

Real-Time Online Learning Trajectory Prediction via Efficient Latent Predictor

Jierui Peng; Department of Computer and Data Science, Case Western Reserve University; Cleveland; OH
Vipin CHaudhary; Department of Computer and Data Science, Case Western Reserve University; Cleveland; OH
Yu Yin; Department of Computer and Data Science, Case Western Reserve University; Cleveland; OH

Abstract

Trajectory prediction is crucial for autonomous systems, but traditional deep learning models, typically trained on specific pre-collected trajectories, often fail to generalize to unseen scenarios due to distribution shifts. Recent approaches address this by integrating online learning for adaptive deployment. However, existing online learning methods face two major challenges: (1) long training times, which prevent real-time execution, and (2) failure to account for variations in input data speed, leading to performance degradation when processing high-speed dynamic scenarios. To overcome these limitations, we introduce a latent-space predictor that forecasts future trajectories by aligning learned latent representations with encoded ground truth. This approach enhances robustness to distribution shifts while reducing reliance on direct coordinate regression. Additionally, we incorporate a lightweight online learning module, enabling efficient real-time adaptation without full model retraining. We evaluate our method on nuScenes, Waymo, and Lyft L5 datasets, focusing on data distribution shift scenarios. Experimental results demonstrate that our model outperforms state-of-the-art online learning methods, achieving approximate 9.9% improvement in trajectory prediction accuracy while significantly reducing optimization time up to 54%.

Introduction

Trajectory prediction is essential for autonomous driving and robotics, enabling agents to anticipate the motion of dynamic objects for safe navigation [1, 2, 3, 4, 5]. Deep learning has significantly advanced this field, utilizing sequence models [1], graph-based networks [6, 7], and Transformers [8, 9] to effectively capture spatial and temporal dependencies. However, most contemporary models are trained offline using a supervised approach, relying on regression loss computed from pre-collected datasets. This standard training paradigm relies on the critical, yet often violated, assumption that the training and real-world deployment data distributions will remain consistent [10]. In practice, data distribution shifts are inevitable, arising from changes in traffic dynamics, sensor noise, and domain variations (*e.g.,* urban vs. rural driving), which severely limit the models' ability to generalize and lead to unreliable motion forecasts in unseen environments.

To address these generalization limits, recent research has focused on online learning or test-time adaptation (TTA) techniques [11, 12, 13, 14, 15], which dynamically refine model predictions based on incoming observations. Despite this progress, a major challenge persists: computational inefficiency in real-time adaptation. Many existing online methods necessitate redundant model structures or rely on full gradient-based optimization at each timestamp, resulting in excessive computation and long optimization times [16, 14]. This inefficiency is particularly problematic for modern autonomous vehicles, which operate at high frame rates (*e.g.,* 30 FPS and higher) using high-resolution sensors. Consequently, the slow adaptation process inherent in traditional online learning methods prevents real-time execution and hinders the reliable deployment of trajectory prediction systems in dynamic, high-speed scenarios.

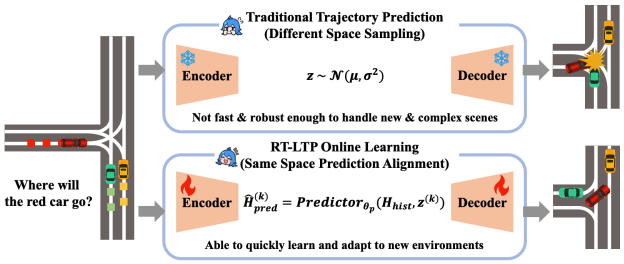


Figure 1. *Comparison between traditional trajectory prediction and the proposed RT-LTP framework. Top: Traditional models sample future trajectories from a separate high-dimensional space, leading to poor generalization under distribution shift. Bottom: RT-LTP directly predicts future embeddings in a consistent latent space, enabling faster and more robust real-time adaptation to new environments.*

The core of the computational inefficiency and generalization failure in online adaptation lies in the fundamental structure of the feature space employed by these models. Many trajectory prediction methods [17, 18] encode historical trajectories into a high-dimensional feature space and then use generative models (*e.g.,* Conditional Variational AutoEncoders [19] or diffusion models [20]) to transform this representation into a future feature space. While this approach enables modeling of complex trajectory distributions within a specific dataset, it inherently assumes that training and test motions follow the same distribution and introduces a feature-space gap between historical and future trajectories, which limits generalization to new scenarios. Meanwhile, this transformation introduces an intrinsic gap between the historical and future feature spaces. These models typically operate under the assumption that training and test motion distributions are aligned. When distributional shifts occur (*e.g.,* in novel or dynamic environments) this pre-existing feature space gap severely hinders generalization. Consequently, the online adaptation mechanism is forced to perform a computationally expensive, non-linear re-alignment of these disparate spaces. This struggle to reconcile inconsistent representations significantly increases optimization time and computational cost, and often leads to degraded predictive performance. To enable efficient real-time

adaptation, it is therefore critical to design models with feature space consistency, ensuring that historical and future representations are inherently aligned and jointly learnable.

To overcome these limitations, we propose RT-LTP (Real-Time Latent Trajectory Predictor), an efficient online learning framework for real-time trajectory forecasting in high-speed autonomous systems. RT-LTP targets two key challenges in test-time deployment: limited generalization under distributional shift and high adaptation cost. As illustrated in Figure 1, traditional methods sample from a future latent space disconnected from the past, leading to misaligned representations. In contrast, RT-LTP reformulates trajectory prediction as a latent-space forecasting task, predicting future motion directly in a compact, semantically structured space shared by both past and future. By aligning predicted latent features with frozen embeddings of ground-truth futures, RT-LTP enforces feature space consistency, which stabilizes adaptation and reduces overfitting. This is particularly beneficial when only partial future observations are available: rather than depending on sparse or noisy coordinate supervision, the model can still rely on the structured latent space to reason about high-level intent (*e.g.,* turning or stopping), making online learning more reliable.

To operate effectively under online learning settings, where models must adapt continually during deployment, The variant of RT-LTP, Fast RT-LTP, introduces a lightweight LoRA-ACT module that enables fast, real-time optimization without retraining the full network. By dynamically updating only a small set of low-rank parameters within the latent predictor's attention layers while keep encoder and decoder fixed, RT-LTP achieves substantial reductions in computational overhead and memory footprint. This design allows the model to adapt on the fly to distribution shifts and dynamic scene changes, maintaining high forecasting accuracy even at high input frequencies typical of real-world autonomous driving. As demonstrated in Figure 2, RT-LTP and Fast RT-LTP delivers a superior trade-off between speed and accuracy, establishing a new standard for scalable, real-time trajectory forecasting.

In summary, our main contributions are as follows:

- We propose a trajectory forecasting framework that reformulates prediction as a latent-space alignment problem. By predicting future motion in a latent space, we reduce reliance on coordinate regression and enables more efficient learning.
- We introduce LoRA-ACT, a *dynamic* low-rank adaptation mechanism that selectively updates a small subset of parameters in the latent predictor during online adaptation, eliminating the need for full model updates and enabling real-time deployment even under high-frequency sensor inputs.
- Our method achieves up to 54% faster optimization and 9.9% higher accuracy, validated across three datasets under distribution shift scenarios, demonstrating superior efficiency and robustness for real-time autonomous applications.

Related Works

Online Learning for Motion Forecasting A key challenge in online learning for trajectory prediction is the high computational cost of real-time adaptation. Many existing methods rely on full gradient-based optimization at test time, making them impractical for real-time deployment. Some mitigate this by updating only specific layers, such as batch normalization [14, 21], but this limits its adaptability to distribution shifts.

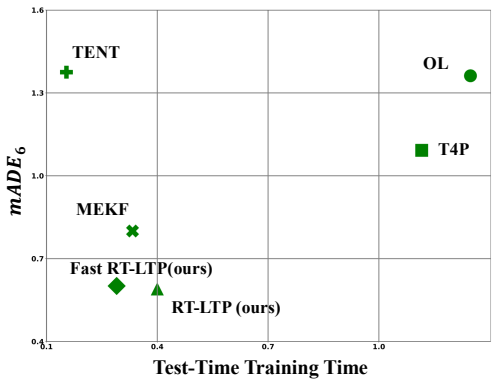


Figure 2. *Comparison of Test-Time Training trajectory forecasting models in terms of prediction error and test-time training time. Each marker represents a model, with lower-left positions indicating better trade-offs between speed and accuracy. The proposed RT-LTP and Fast RT-LTP achieve the best balance between efficiency and precision, outperforming prior methods.*

This issue worsens in autonomous vehicles, where high-resolution sensors operate at varying FPS rates. Most trajectory forecasting models assume a fixed, low input speed, making them inefficient in high-speed scenarios. As the number of input frames per each time unit increases, adaptation methods [11, 16] struggle to process data efficiently, leading to delayed predictions and degraded accuracy—critical for real-time motion forecasting.

To address this, we integrate an efficient latent-space adaptation strategy with a lightweight online learning module, reducing computational costs while preserving adaptability. Our model is explicitly designed for high-speed input processing, ensuring stable performance in dynamic driving environments.

Latent-space Learning Unlike traditional deterministic models, latent-space learning methods encode trajectory dynamics into a low-dimensional latent space [22, 23], allowing for more structured and flexible representations. Recent works explore various ways to structure latent spaces. [23] employs self-supervised masked auto-encoding, while [24] introduces orthogonal latent bases to encode meaningful motion semantics, improving interpretability and controllability.

Another crucial benefit of latent-space learning is its efficiency in online adaptation and real-time prediction. Traditional gradient-based optimization methods for online learning are computationally expensive, making them impractical for real-time deployment. Latent-space representations allow models to adapt without extensive gradient updates [25, 22].

Our approach extends this framework by integrating a latent predictor trained with embedded ground truth, enabling efficient online adaptation and robustness to distribution shifts.

Method

To enable robust and efficient trajectory forecasting in real-world scenarios, we propose the **Real-Time Latent Trajectory Predictor (RT-LTP)**, a novel framework designed for fast, test-time adaptation via latent-space forecasting. Unlike conventional models that rely on direct coordinate regression, which can be sensitive to noise and computationally expensive to adapt online,

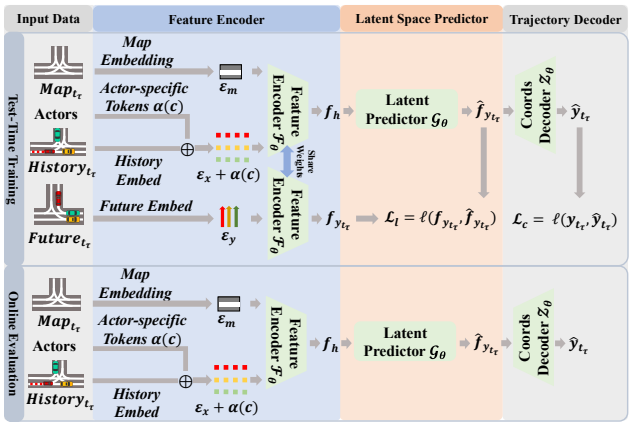


Figure 3. Real-Time Latent Trajectory Predictor (RT-LTP). *The model processes map features and historical trajectories through a feature encoder with actor-specific tokens, while future supervision signals are embedded separately using a different encoder. The encoded history and map features are passed through the latent predictor, which generates future latent representations. The pipeline is supervised by the encoded and coordinates-level future trajectories, ensuring adaptation through latent-space loss ($\mathscr{L}_l$) and coordinate-space loss ($\mathscr{L}_c$).*

RT-LTP predicts future motion in a compact, semantically structured latent space. RT-LTP offers a lightweight, real-time forecasting solution that significantly improves adaptation robustness and efficiency. In the remainder of this section, we formalize the problem setup and real-time adaptation challenges (Sec.), introduce our latent-space prediction architecture (Sec.), and describe the online learning mechanism that enables efficient test-time updates (Sec.).

Problem Definition

Mathematical Definition The objective is to predict the joint future trajectories of N agents $\mathscr{A} = \{a_1, \ldots, a_N\}$, where a_1 is the ego vehicle. For each agent a_i, we define its historical trajectory over T_h timesteps as $X^i = \{x_t^i \in R^d \mid t = -T_h, \ldots, 0\}$ and its future trajectory over T_f steps as $Y^i = \{y_t^i \in R^d \mid t = 1, \ldots, T_f\}$. Given the full set of histories $X^{1:N}$ and the environmental map context $\mathscr{M}$ (e.g., lane topology and connectivity), we learn a function $\mathscr{F}_\theta$ to produce joint forecasts:

$$\hat{Y}^{1:N} = \mathscr{F}_\theta(X^{1:N}, \mathscr{M}) \tag{1}$$

The model $\mathscr{F}_\theta$ utilizes attention mechanisms to encode multi-agent interactions and spatial relationships between agents and the map context.

Online Learning with Partial Future Observations We assume partial ground-truth future trajectories $Y_{\text{obs}}^i = \{y_t^i \in Y^i \mid t \le T_o\}$ become observable after a short delay, where $T_o < T_f$. These observations provide a weak supervision signal for real-time adaptation. At each timestamp t_τ, aligned with an optimization stride τ, the model receives new partial observations $Y_{\text{obs}}^{1:N}(t_\tau)$ to refine its predictions. The resulting adapted prediction function is:

$$\hat{Y}'^{1:N} = \mathscr{F}'_\theta(X^{1:N}, \mathscr{M}, Y_{\text{obs}}^{1:N}) \tag{2}$$

where $\mathscr{F}'_\theta$ denotes the model parameters updated via online learning.

Real-Time Constraint To ensure that our method is deployable in real-world autonomous systems, we impose a strict real-time constraint: the total runtime, including both online adaptation and inference, must not exceed 1 second. This threshold follows prior works on low-latency adaptation [26, 27], and is designed as a conservative upper bound on per-update computation time, rather than a delay in reaction. In practice, our model operates well within this limit. For example, **Fast RT-LTP** achieves online update and inference in approximately 0.33 seconds per cycle, leaving substantial time for downstream planning and control.

Optimization Stride To simulate the diverse update rates encountered in real-world sensor systems, we define the **optimization stride** as the number of input frames processed between two consecutive online adaptation steps. This stride directly controls how frequently the model updates its parameters during deployment, reflecting the latency and bandwidth constraints of the sensing pipeline. For example, a stride of 30 indicates that the model performs one adaptation step after processing 30 frames. This is equivalent to once per second at 30 FPS. Smaller strides (*e.g.,* 12) correspond to more frequent updates and faster adaptation, while larger strides (*e.g.,* 60) mimic lower-frequency updates typical of bandwidth-limited or delayed systems.

Latent-Space Predictor

Our framework consists of three components: latent representation learning, latent future prediction, and trajectory decoding. The latent predictor is defined as:

$$\hat{Y}_l = \mathscr{G}_\theta(\mathscr{F}_\theta(X, \mathscr{M})) \tag{3}$$

where $\mathscr{F}_\theta$ is a shared Transformer-based encoder for historical motion X and map context $\mathscr{M}$, and $\mathscr{G}_\theta$ is a latent predictor. To capture agent-specific behavior, we add learnable actor tokens $\alpha(c)$ to the trajectory embeddings: $h_x = E(X) + \alpha(c)$ and $h_m = E(\mathscr{M})$.

To train $\mathscr{G}_\theta$, we align its prediction with a target embedding Y_l obtained by passing the ground-truth future Y through the encoder at time t_τ using a `stopgrad` operation:

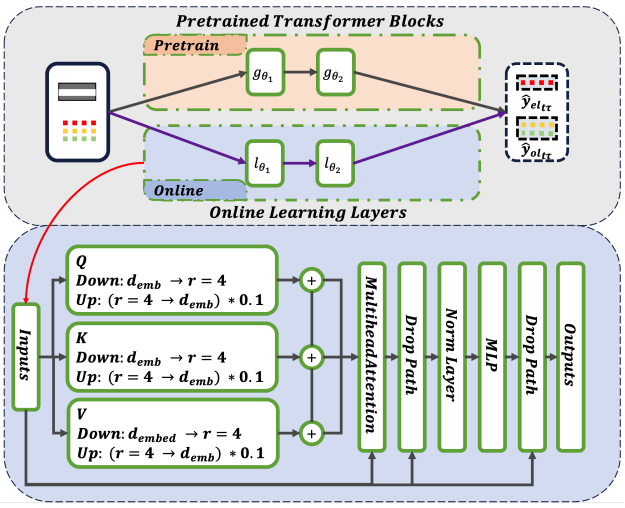
$$Y_l = \texttt{stopgrad}(\mathscr{F}_\theta(Y_{t_\tau})) \tag{4}$$

This design, inspired by I-JEPA [28], ensures a stable supervision target and forces the model to learn meaningful temporal mappings. We supervise the ego and surrounding agents separately via a factorized latent alignment loss:

$$\mathscr{L}_l = \lambda_e \cdot \ell_1(\hat{Y}_{el}, Y_{el}) + \lambda_o \cdot \ell_1(\hat{Y}_{ol}, Y_{ol}) \tag{5}$$

This complements the coordinate-space loss $\mathscr{L}_c = \frac{1}{N}\sum_{n=1}^{N} \min_{k \in K} \|y_{t_\tau}^n - \hat{y}_{t_\tau}^{n,k}\|^2$. While $\mathscr{L}_c$ supervises physical accuracy, $\mathscr{L}_l$ enforces semantic consistency and provides robustness to sensor noise and sparse supervision during adaptation.

Compared to prior works [11, 29], our method bypasses heavy reconstruction or contrastive objectives. By directly aligning predicted trajectories with frozen embeddings, we achieve a compact, expressive, and end-to-end forecasting pipeline tailored for real-time deployment.



justification=justified, singlelinecheck=false

Figure 4. ***Efficient Latent Predictor****. In pretraining, the data passes through the regular Transformer blocks ($g_{\theta i}$ from $\mathscr{G}_\theta$), where the full model parameters are optimized to achieve high accuracy without time constraints. During online learning, the data is processed through the online learning layers $l_{\theta i}$, where a learnable low-rank matrix is introduced to reduce the number of trainable parameters by approximately 70%. The latent predictor produces latent representations for both the ego agent and other agents.*

Online Learning Process

As the system advances, partial ground-truth future segments Y_{obs} become observable. We use these points as weak supervision for periodic online adaptation based on an optimization stride. During adaptation, the encoder $\mathscr{F}_\theta$ and decoder $\mathscr{L}_\theta$ are frozen. We generate a latent supervision signal from the observed future:

$$Y_l^{\text{obs}} = \texttt{stopgrad}(\mathscr{F}_\theta(Y_{\text{obs}})) \tag{6}$$

The latent predictor is updated by minimizing a joint loss: $\mathscr{L} = \lambda_l \mathscr{L}_l + \lambda_c \mathscr{L}_c$, where $\mathscr{L}_l$ enforces latent consistency and $\mathscr{L}_c$ provides coordinate-space supervision via a Winner-Takes-All (WTA) loss.

LoRA-ACT : To enable efficient adaptation, we propose *LoRA-ACT*, which injects trainable low-rank matrices into the Q, K, V projections of the latent predictor. We dynamically adjust the rank r based on the change in a feedback metric $\mathscr{C}_t$ ($mADE_6$). Starting at $r = 4$, the rank expands if performance degrades significantly:

$$r_t = \begin{cases} 4, & \text{initialization} \\ \min(0.1 d_{\text{emb}}, r_{t-1} + 1), & \text{if } \mathscr{C}_t - \mathscr{C}_{t-1} > \delta_{\text{inc}} \\ r_{t-1}, & \text{otherwise} \end{cases} \tag{7}$$

This mechanism allows the model to increase capacity during domain shifts while maintaining a lightweight profile for real-time deployment. By updating only the low-rank parameters, LoRA-ACT preserves predictive accuracy across diverse environments without high computational overhead.

Experiments

Setup

We evaluate our approach on four motion forecasting benchmarks: **INTERACTION** [30], **nuScenes** [31], **Waymo Open Motion** [32], and **Lyft Level 5** [33]. Using the **TrajData** framework [34], we utilize a 2.0s historical context and 6.0s prediction horizon sampled at 2Hz (input/output lengths of 5/12).

Baselines include our **Pretrained** (non-adaptive) backbone, **ALPaCA** [35], **TENT** [36], **MEKFλ** [37], **OL** [14], and **T4P** [11]. To assess performance under varying temporal constraints, we evaluate across multiple optimization strides (12, 24, 25, 30, and 60), representing update rates from 2.5 Hz to 0.5 Hz. This range reflects typical sensor frequencies and tests the model's robustness to infrequent updates.

Results

Quantitative Results Table 1 summarizes performance across various dataset shifts (Source →Target). Our full model, *RT-LTP*, consistently achieves the highest forecasting accuracy, outperforming both offline baselines and existing online adaptation methods. This performance highlights the model's robustness to domain shift and multimodal uncertainty.

Central to this success is our design choice to perform trajectory prediction within a structured latent space. By encoding agent histories and map context into a shared, temporally-aligned representation, *RT-LTP* ensures that prediction and supervision operate within the same feature space, avoiding distribution mismatches inherent in coordinate-space regression. Aligning the predictor with frozen ground-truth future embeddings—acting as semantic anchors—enables the model to learn domain-invariant motion patterns. Consequently, *RT-LTP* maintains high-fidelity predictions in complex scenes involving abrupt maneuvers and dense interactions, bridging the training-deployment gap.

Efficiency While RT-LTP offers the best accuracy, *Fast RT-LTP* can significantly reduce the number of parameters involved in online learning. As shown in Figure 6, Fast RT-LTP maintains strong predictive performance with only a marginal drop in accuracy compared to RT-LTP (see Table 1), but achieves up to *54% reduction in total adaptation time*. Notably, both RT-LTP and Fast RT-LTP remain well below the 1-second real-time latency threshold even as the optimization stride increases. This ensures seamless integration with high-frequency sensor inputs and guarantees timely predictions in safety-critical autonomous systems. Our framework therefore offers a practical trade-off: RT-LTP for maximum accuracy, and Fast RT-LTP for latency-sensitive deployments, both outperforming existing methods.

Short Term Prediction In the short-term forecasting setup, following the T4P definition, models are required to predict 3 seconds into the future given only 0.9 seconds of past observations sampled at 0.1s intervals, resulting in input and output sequence lengths of 10 and 30, respectively. This configuration imposes a more extreme prediction challenge with limited historical context and rapid temporal dynamics. As shown in Table 2, *RT-LTP* consistently achieves the lowest mADE_6 and mFDE_6 across all cross-dataset shifts, demonstrating strong generalization and robustness even under highly constrained conditions. By contrast, *Fast RT-LTP* exhibits a slight degradation in accuracy due to its LoRA-based lightweight adaptation. These results underscore

Table 1. Each model is trained on the source dataset and performs online learning on the target dataset (source → target). We evaluate performance using $mADE_6$ and $mFDE_6$, where lower values indicate better accuracy. Colors indicate the best and second best results respectively.

		nus → lyft		way → lyft		way → nus		Mean	
Models	Online Learning	mADE_6	mFDE_6	mADE_6	mFDE_6	mADE_6	mFDE_6	mADE_6	mFDE_6
Pretrained	✗	0.939	2.311	0.636	1.432	1.75	3.35	1.108	2.364
TENT	✓	1.068	2.514	0.628	1.381	1.077	2.012	0.924	1.969
MEKF	✓	1.006	2.369	0.615	1.351	1.117	2.14	0.913	1.953
ALPaCA	✓	1.462	2.573	0.977	2.184	1.495	2.978	1.311	2.578
OL	✓	1.309	2.484	0.647	1.233	1.123	2.155	1.026	1.957
T4P	✓	0.776	1.820	0.549	1.171	0.996	1.784	0.774	1.592
RT-LTP (ours)	✓	0.546	1.246	0.512	1.167	1.034	2.13	0.697	1.514
Fast RT-LTP (ours)	✓	0.571	1.335	0.513	1.168	1.093	2.303	0.726	1.602

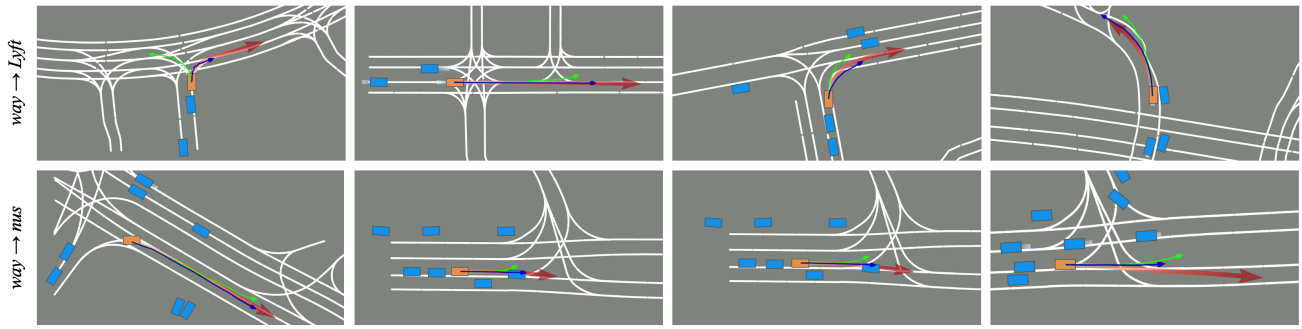


Figure 5. *The figure illustrates ground-truth trajectories (red arrows), our proposed method (RT-LTP) predictions (blue arrows), and baseline T4P predictions (green arrows) across multiple driving scenarios. The white lanes represent driveway on the map. Blue boxes represent other actors(other vehicles, bicycles, pedestrian and motorcycles.) For simplicity, we only depict the best results for multi-modal predictions that closet to the ground-truth trajectories*

Table 2. Comparison of short-term trajectory forecasting performance across cross-dataset distribution shifts. Colors indicate the best and second best results respectively.

Models	INTER ->nus		INTER ->Lyft		nus ->Waymo	
	mADE_6	mFDE_6	mADE_6	mFDE_6	mADE_6	mFDE_6
T4P	0.537	1.137	0.391	0.824	0.336	0.807
RT-LTP (ours)	0.517	1.103	0.358	0.816	0.265	0.679
Fast RT-LTP (ours)	0.543	1.147	0.544	1.275	0.320	0.774

Table 3. RT-LTP performance under edge-case filtering on the Waymo dataset with Lyft pretraining (Lyft → Waymo).

Only Edge	Short Term		Long Term	
Case	$mADE_6$	$mFDE_6$	$mADE_6$	$mFDE_6$
✗	0.265	0.679	1.356	3.401
✓	0.266	0.694	1.408	3.526

RT-LTP's ability to deliver state-of-the-art accuracy under short-horizon prediction tasks, while Fast RT-LTP provides a practical speed–accuracy trade-off for latency-sensitive deployments.

Edge Cases To assess the robustness of RT-LTP under high-risk traffic scenarios, we conduct an evaluation on the Waymo dataset using a model pretrained on Lyft (*i.e.,* Lyft → Waymo). Edge cases are isolated using the TrajData [34] distance filter, selecting scenes where the ego vehicle is within 5 meters of another agent.

Table 4. Impact of different components in our framework on forecasting performance under the long term nus → Lyft setting. The *Predictor* column corresponds to the latent predictor, *Online* indicates the online learning module, and *Efficient* denotes our efficiency-oriented optimization strategy.

Predictor	Modules: Online	Modules: Efficient	$mADE_6$ / $mFDE_6$	Max Optimization Time (s)	Max Inference Time (s)
✓		✓	0.939 / 2.311	N/A	0.25
	✓	✓	0.723 / 1.594	0.17	0.27
✓	✓		0.546 / 1.246	0.54	0.29
✓	✓	✓	0.571 / 1.335	0.33	0.24

We follow the standard short-term (0.1s interval) and long-term (0.5s interval) prediction settings from prior setting. As summarized in Table 3, RT-LTP maintains stable performance in these challenging scenarios, with only minor increases in error metrics. This indicates the model's strong generalization to dense, interactive environments and confirms its reliability under distribution shift and real-world edge conditions.

Ablation Studies

To assess the contributions of each component in our proposed framework, we conduct an extensive ablation study evaluating the impact of the latent predictor, online learning module, efficiency-driven optimization strategy, the number of predictor blocks, and multiple update passes.

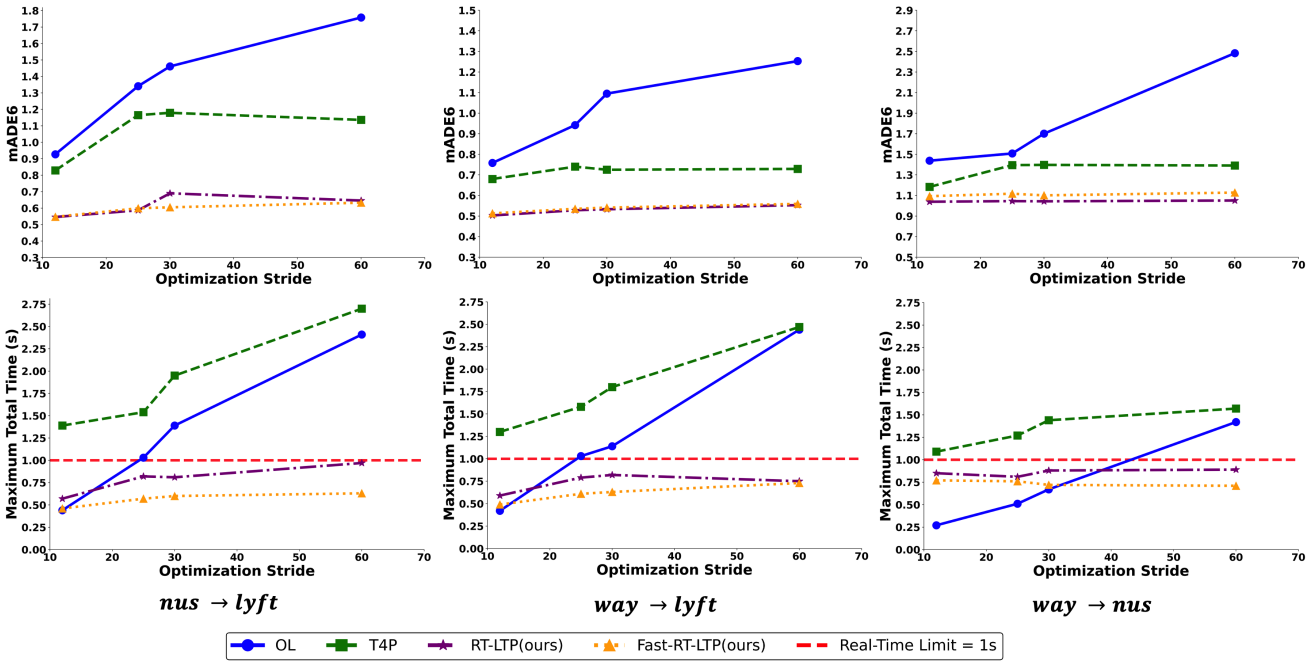


Figure 6. *The top row illustrates the forecasting error as the optimization stride increases, while the bottom row shows the maximum total runtime (including both optimization and inference). Our proposed methods maintain consistently low error while ensuring real-time efficiency, staying within the 1-second threshold (dashed red line).*

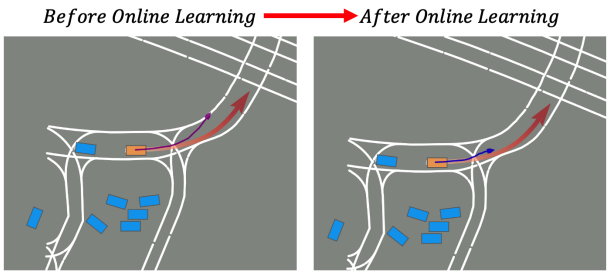


Figure 7. ***Ablation Study on the Online Learning Module.*** *The figure compares trajectory predictions before and after applying the online learning module. The left panel displays the initial prediction (purple arrow) before adaptation, while the right panel shows the refined prediction (blue arrow) after the online learning update.*

Modules Table 4 shows the impact of each module on performance. The latent predictor enhances trajectory accuracy by structuring future motion representations, reducing error by 23%. The online learning module further improves adaptation to distribution shifts, refining predictions as shown in Figure 7. The efficiency module reduces optimization time from 0.54s to 0.33s without compromising accuracy. These results validate our approach's ability to balance computational cost and prediction quality, ensuring real-time adaptability in high-speed autonomous scenarios.

Predictor Blocks Table 5 examines the effect of varying the number of Transformer blocks in the latent predictor. Increasing the number of blocks can possibly improves accuracy up to a certain extent; however, adding more blocks beyond two leads to diminishing returns and increased optimization time. Notably, using two predictor blocks achieves the best trade-off between accuracy and efficiency, reducing mADE_6 from 0.610 to 0.569 while keeping the optimization time within acceptable limits.

Table 5. Ablation study on the latent predictor, showing the effect of varying the number of Transformer blocks on long-term trajectory forecasting performance under the nus → Lyft.

Numbers of Predictor Blocks	mADE_6 / mFDE_6	Max Optimization Time (s)	Max Inference Time (s)
1	0.610 / 1.455	0.39	0.28
2	0.546 / 1.246	0.54	0.27
3	0.647 / 1.533	0.58	0.29
4	0.581 / 1.361	0.61	0.45

Conclusion

In this work, we present RT-LTP, an efficient online learning framework for real-time trajectory prediction. By aligning predicted and ground-truth embeddings in a compact latent space, RT-LTP enables fast, scalable test-time learning without full model updates. Experiments show up to 21% accuracy gains and 54% faster adaptation, demonstrating strong performance under time-sensitive, real-world conditions.

References

[1] L. Lin, W. Li, H. Bi, and L. Qin, "Vehicle trajectory prediction using lstms with spatial–temporal attention mechanisms," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 2, pp. 197–208, 2022.

[2] J. Hagenus, F. B. Mathiesen, J. F. Schumann, and A. Zgonnikov, "Robustness in trajectory prediction for autonomous vehicles: a survey," in *2024 IEEE Intelligent Vehicles Symposium (IV)*, 2024, pp. 969–976.

[3] A. Bighashdel and G. Dubbelman, "A survey on path prediction techniques for vulnerable road users: From traditional to deep-learning approaches," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 1039–1046.

[4] X. Zhang, X. Fu, Z. Xiao, H. Xu, and Z. Qin, "Vessel trajectory prediction in maritime transportation: Current approaches and beyond," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 19 980–19 998, 2022.

[5] Z. Zhou, J. Wang, Y.-H. Li, and Y.-K. Huang, "Query-centric trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 863–17 873.

[6] Y. Xu, L. Wang, Y. Wang, and Y. Fu, "Adaptive trajectory prediction via transferable gnn," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 6520–6531.

[7] T. Westny, J. Oskarsson, B. Olofsson, and E. Frisk, "Mtp-go: Graph-based probabilistic multi-agent trajectory prediction with neural odes," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 9, pp. 4223–4236, 2023.

[8] F. Giuliari, I. Hasan, M. Cristani, and F. Galasso, "Transformer networks for trajectory forecasting," in *2020 25th international conference on pattern recognition (ICPR)*. IEEE, 2021, pp. 10 335–10 342.

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

[10] T. Phong, H. Wu, C. Yu, P. Cai, S. Zheng, and D. Hsu, "What truly matters in trajectory prediction for autonomous driving?" *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[11] D. Park, J. Jeong, S.-H. Yoon, J. Jeong, and K.-J. Yoon, "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. 15 065–15 076.

[12] B. Yang, J. Zhu, Z. Yu, F. Fan, X. Liu, and R. Ni, "Fast adaptation trajectory prediction method based on online multi-source transfer learning," *IEEE Transactions on Automation Science and Engineering*, 2024.

[13] P. Yao, T. Mao, M. Shi, J. Sun, and Z. Wang, "Eanet: Expert attention network for online trajectory prediction," *arXiv preprint arXiv:2309.05683*, 2023.

[14] C. Hao, Y. Chen, S. Cheng, and H. Zhang, "Improving vehicle trajectory prediction with online learning," in *2023 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2023, pp. 1–7.

[15] M. Huynh and G. Alaghband, "Aol: Adaptive online learning for human trajectory prediction in dynamic video scenes," *arXiv preprint arXiv:2002.06666*, 2020.

[16] X. Sun, H. Sun, B. Li, D. Wei, W. Li, and J. Lu, "Moml: Online meta adaptation for 3d human motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 1042–1051.

[17] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *European Conference on Computer Vision*. Springer, 2020, pp. 683–700.

[18] T. Gu, G. Chen, J. Li, C. Lin, Y. Rao, J. Zhou, and J. Lu, "Stochastic trajectory prediction via motion indeterminacy diffusion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 17 113–17 122.

[19] A. Shocher, A. Dravid, Y. Gandelsman, I. Mosseri, M. Rubinstein, and A. A. Efros, "Idempotent generative network," *arXiv preprint arXiv:2311.01462*, 2023.

[20] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in neural information processing systems*, vol. 28, 2015.

[21] Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros, and M. Hardt, "Test-time training with self-supervision for generalization under distribution shifts," in *International conference on machine learning*. PMLR, 2020, pp. 9229–9248.

[22] R. Benaglia, A. Porrello, P. Buzzega, S. Calderara, and R. Cucchiara, "Trajectory forecasting through low-rank adaptation of discrete latent codes," in *International Conference on Pattern Recognition*. Springer, 2024, pp. 236–251.

[23] R. Wagner, O. S. Tas, M. Klemp, and C. Fernandez, "Joint-motion: Joint self-supervision for joint motion prediction," *arXiv preprint arXiv:2403.05489*, 2024.

[24] G. Xu, J. Tao, W. Li, and L. Duan, "Learning semantic latent directions for accurate and controllable human motion prediction," in *European Conference on Computer Vision*. Springer, 2024, pp. 56–73.

[25] N. Thakkar, K. Mangalam, A. Bajcsy, and J. Malik, "Adaptive human trajectory prediction via latent corridors," in *European Conference on Computer Vision*. Springer, 2024, pp. 297–314.

[26] H. A. Al Kader Hammoud, A. Prabhu, S.-N. Lim, P. H. Torr, A. Bibi, and B. Ghanem, "Rapid adaptation in online continual learning: Are we evaluating it right?" in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 18 852–18 861.

[27] Y. Ghunaim, A. Bibi, K. Alhamoud, M. Alfarra, H. A. Al Kader Hammoud, A. Prabhu, P. H. Torr, and B. Ghanem, "Real-time evaluation in online continual learning: A new hope," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 11 888–11 897.

[28] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, and N. Ballas, "Self-supervised learning from images with a joint-embedding predictive architecture," 2023. [Online]. Available: https://arxiv.org/abs/2301.08243

[29] Y. Wang, P. Zhang, L. Bai, and J. Xue, "Fend: A future enhanced distribution-aware contrastive learning

framework for long-tail trajectory prediction," 2023. [Online]. Available: https://arxiv.org/abs/2303.16574

[30] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle *et al.*, "Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps," *arXiv preprint arXiv:1910.03088*, 2019.

[31] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[32] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454.

[33] J. Houston, G. Zuidhof, L. Bergamini, Y. Ye, L. Chen, A. Jain, S. Omari, V. Iglovikov, and P. Ondruska, "One thousand and one hours: Self-driving motion prediction dataset," in *Conference on Robot Learning*. PMLR, 2021, pp. 409–418.

[34] B. Ivanovic, G. Song, I. Gilitschenski, and M. Pavone, "trajdata: A unified interface to multiple human trajectory datasets," *Advances in Neural Information Processing Systems*, vol. 36, pp. 27 582–27 593, 2023.

[35] B. Ivanovic, J. Harrison, and M. Pavone, "Expanding the deployment envelope of behavior prediction via adaptive meta-learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7786–7793.

[36] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell, "Tent: Fully test-time adaptation by entropy minimization," *arXiv preprint arXiv:2006.10726*, 2020.

[37] L. Wang, Y. Hu, and C. Liu, "Online adaptation of neural network models by modified extended kalman filter for customizable and transferable driving behavior prediction," *arXiv preprint arXiv:2112.06129*, 2021.

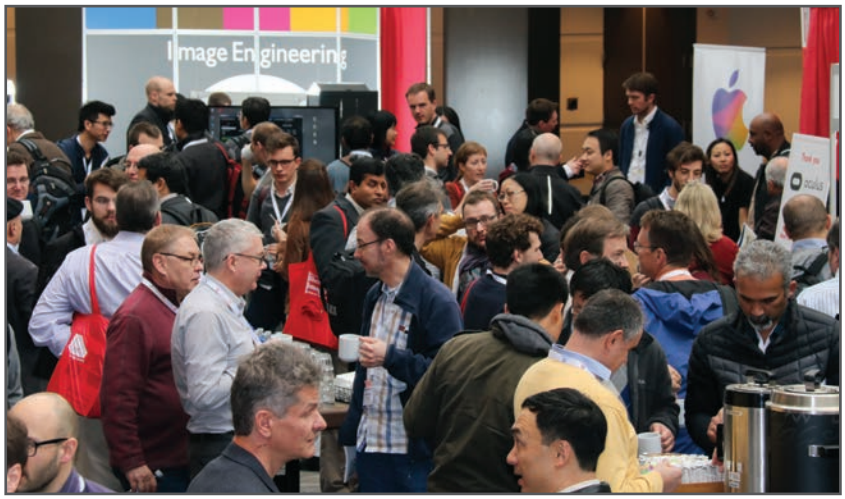
Author Biography

Jierui Peng received his BS from Brandeis University and his Master's degree from New York University (NYU). He is currently pursuing a PhD at Case Western Reserve University (CWRU). He serves as a Research Assistant at both CWRU and University Hospitals (UH). His research focuses on Embodied AI, latent learning, and online learning..

JOIN US AT THE NEXT EI!

electronic IMAGING

Imaging across applications . . . Where industry and academia meet!



• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

ADVANCING IMAGING SCIENCE 75 1947 IS&T 2022 imaging.org