IEEE GLOBECOM®

Segment-Wise Flow Matching for Vision-Aided mmWaveV2I Beam Prediction

Can Zheng, Jiguang He, Chung G. Kang, Guofa Cai, Chongwen Huang, Henk Wymeersch

Korea University, Seoul, Republic of Korea
Great Bay University, Dongguan, China
Guangdong University of Technology, Guangzhou, China
Zhejiang University, Hangzhou, China
Chalmers University of Technology, Gothenburg, Sweden

Presenter: Can Zheng
Date: 2026/12/xx

Contents

01 Background and Motivation

02 System Model and Problem Formulation

03 Proposed Method

04 Simulation Results

05 Conclusions

Contents

01 Background and Motivation

02 System Model and Problem Formulation

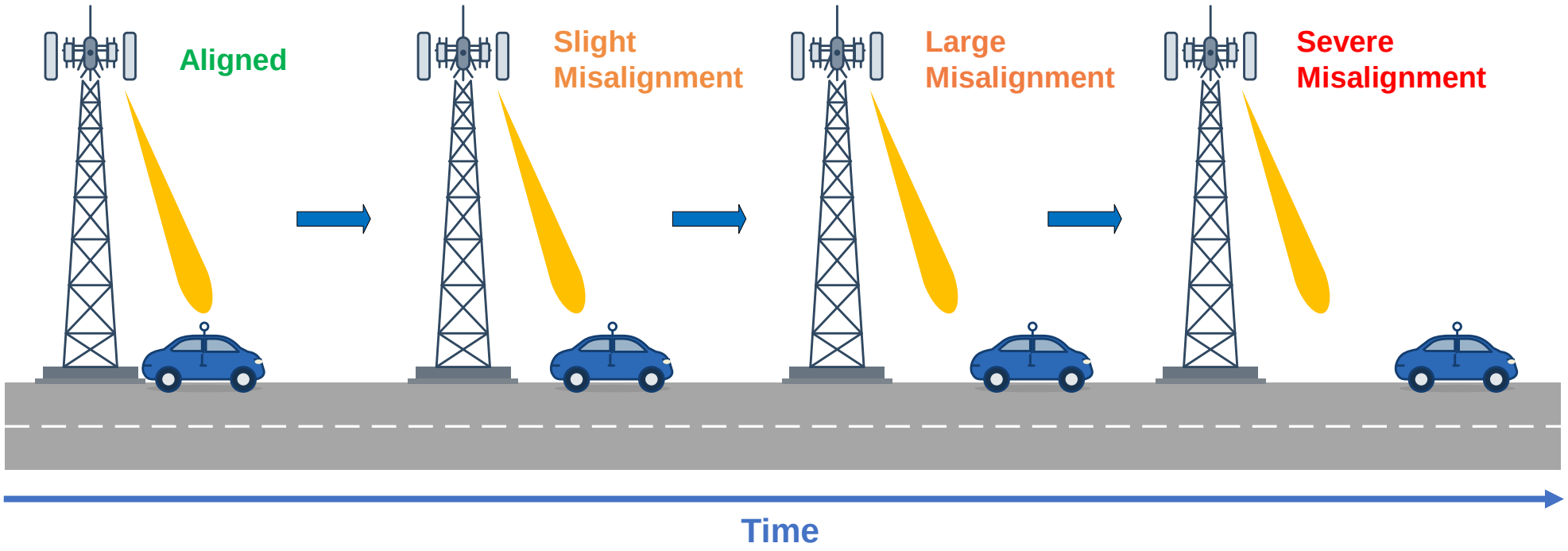
03 Proposed Method

04 Simulation Results

05 Conclusions

Why Do we Need Beam Prediction?

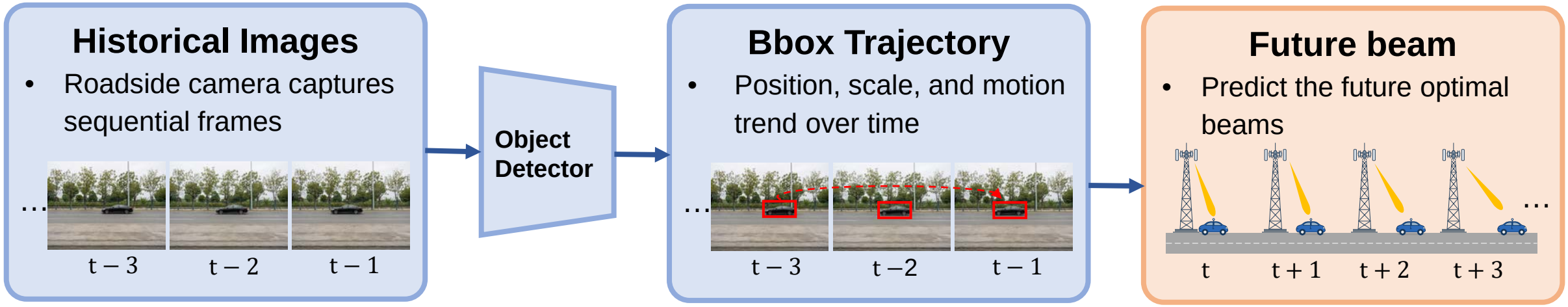
- mmWave can provide **high data rates**, but narrow beams are **highly sensitive** to direction changes.
- High mobility, road obstructions, and environmental variations cause frequent **beam misalignment**.
- Repeated beam sweeping and training increase **overhead** and **latency**.



Key idea: High mobility scenarios require the beam to be switched proactively before the channel changes.

Predict Future Beams from Visual Information

- A monocular camera on the roadside unit (RSU) continuously observes the road and the target vehicle.
- The target vehicle's **bounding box (bbox)** sequence reflects its position, scale, and motion trend.
- The historical visual trajectory can be used to predict the current and future optimal mmWave beams.



Challenges

Complex Mapping

- Visual trajectory -> wireless channel -> optimal beam index
- Nonlinear, environment-dependent

Temporal Consistency

- Coherent sequence
- Not isolated beam index decisions

Low Latency

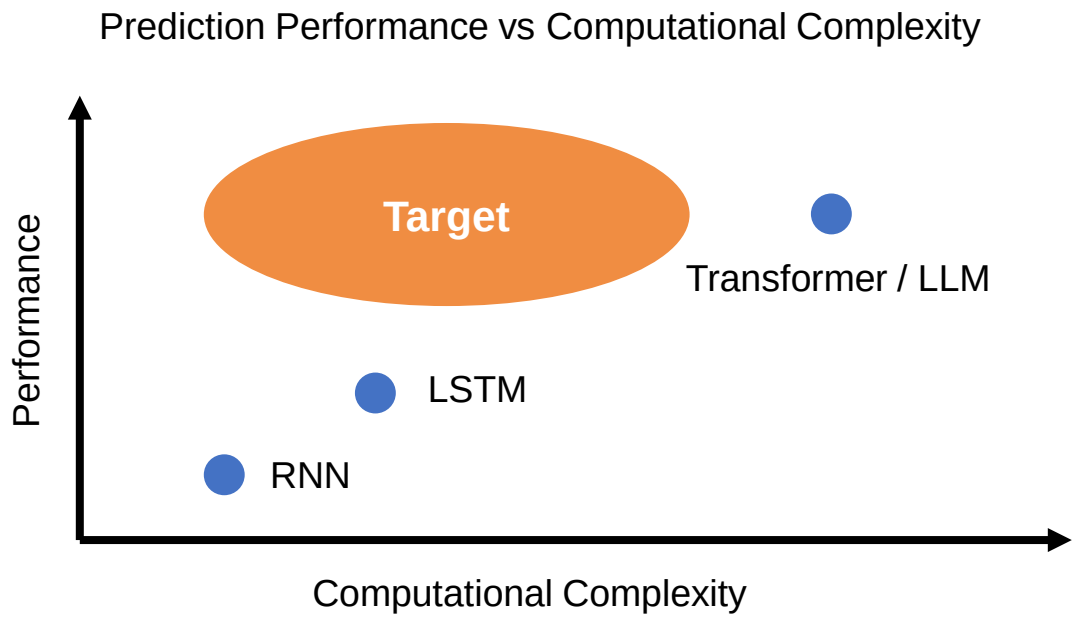
- Edge inference requirement
- Channel variation

Existing Prediction Method

RNN
- √ Lightweight
- × Weak long-term modeling

LSTM
- √ More stable than RNN
- × Weak long-term modeling

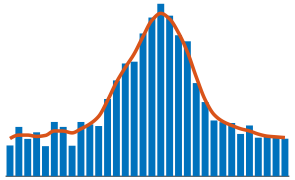
Transformer/LLM
- √ Strong representation
- × Large #parameter & inference time



Can we achieve high accuracy and low inference latency at the same time?

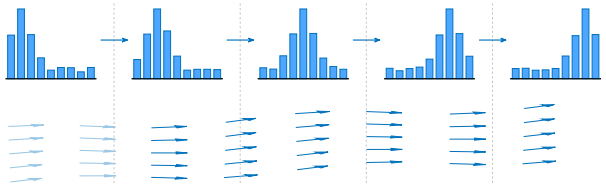
Main Contributions

Beam Power Domain Modeling



- Model beam evolution in the **power domain** over a predefined codebook.
- Avoid directly fitting a discrete beam index sequence.

Segment-wise Flow Matching



- Construct **local flows** between adjacent beam states.
- Learn step-by-step evolution velocity to strengthen temporal consistency.

Efficient Inference



- Use only a few **Euler integration** steps to obtain the prediction sequence.
- Significantly reduce inference cost while maintaining accuracy.

Accuracy, Consistency, and Efficiency.

Contents

01 Background and Motivation

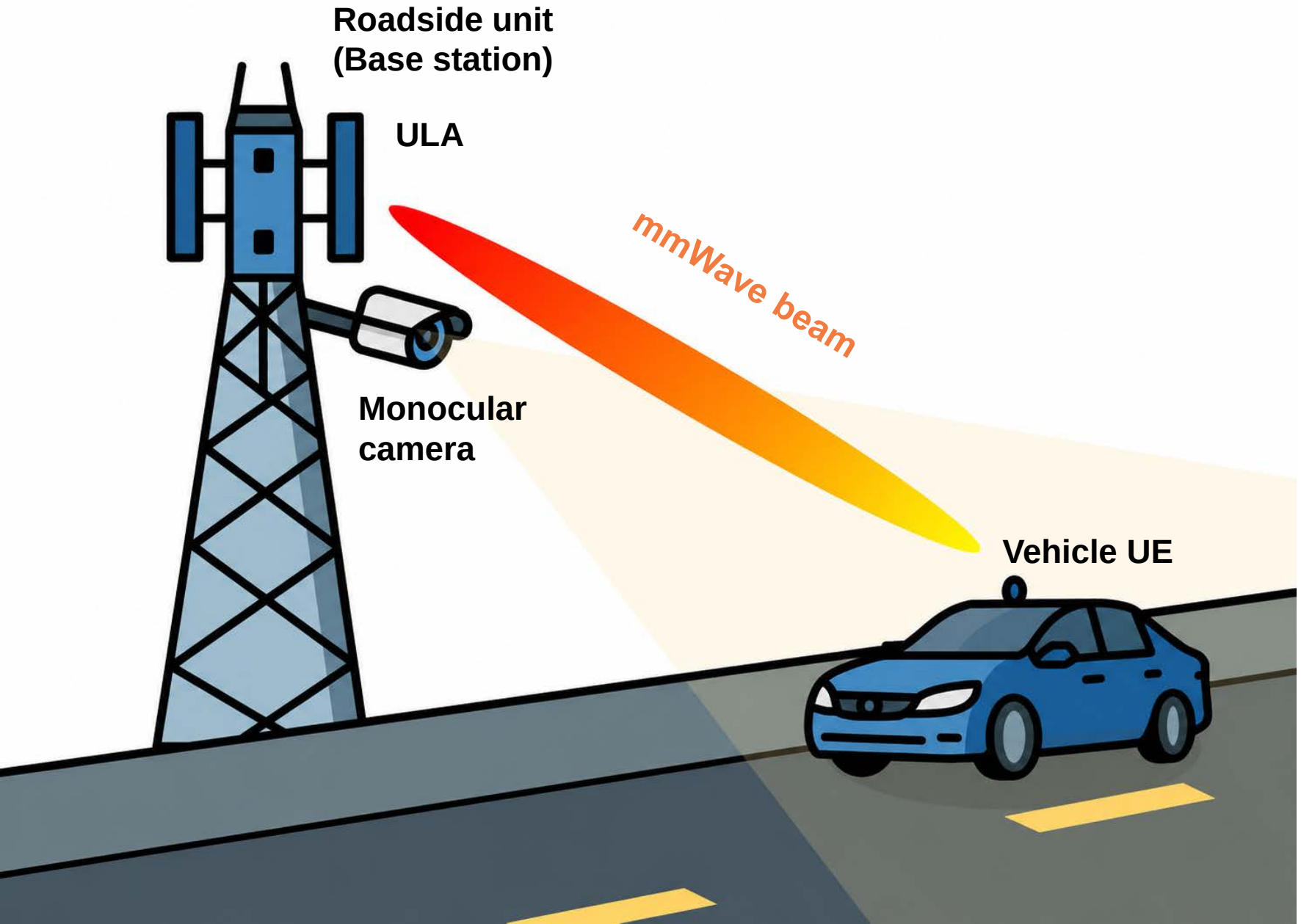
02 System Model and Problem Formulation

03 Proposed Method

04 Simulation Results

05 Conclusions

Scenario



Roadside unit

N-element ULA

Single RF chain
Analog beamforming

Beam Codebook

M fixed candidate beams

$$\mathcal{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_M\}$$

Camera

Monocular RGB camera
Sampling rate: approx. 10 Hz

User Equipment

Single-antenna vehicle user

System Model

1. Downlink received signal

$$r_t = \mathbf{h}_t^{\mathsf{H}} \mathbf{w}_m s_t + n_t$$

Channel, beamformer, transmitted symbol, and noise

2. Received power over the codebook

$$\mathbf{p}_t = [|\mathbf{h}_t^{\mathsf{H}} \mathbf{w}_1|^2, |\mathbf{h}_t^{\mathsf{H}} \mathbf{w}_2|^2, \ldots, |\mathbf{h}_t^{\mathsf{H}} \mathbf{w}_M|^2]^{\mathsf{T}}$$

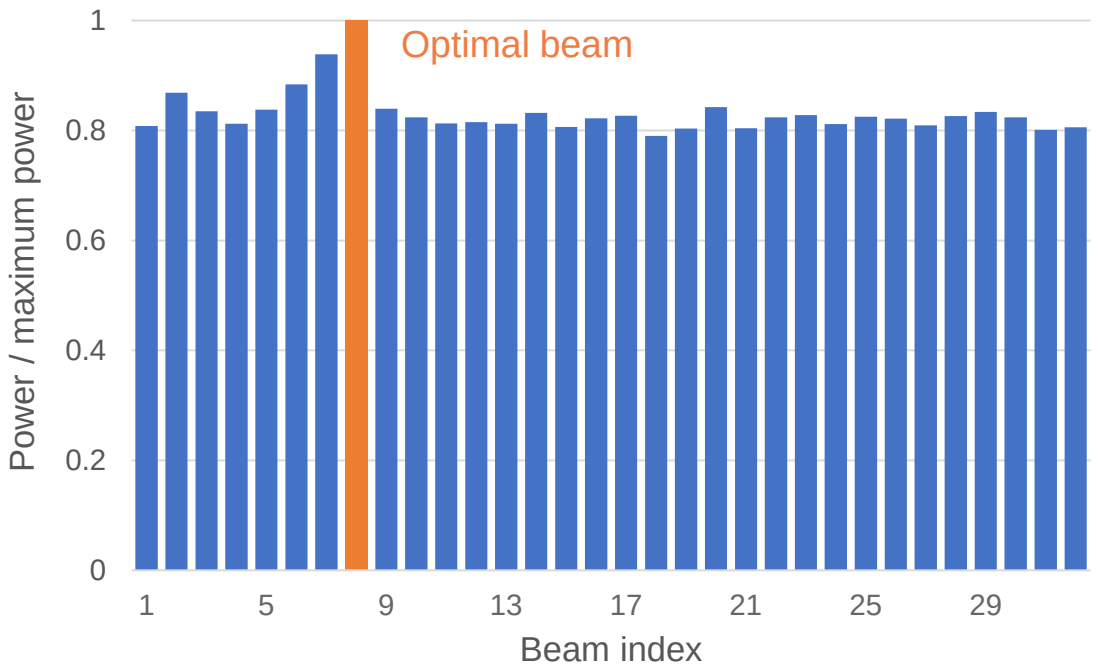
One power value for every candidate beam.

3. Optimal beam index

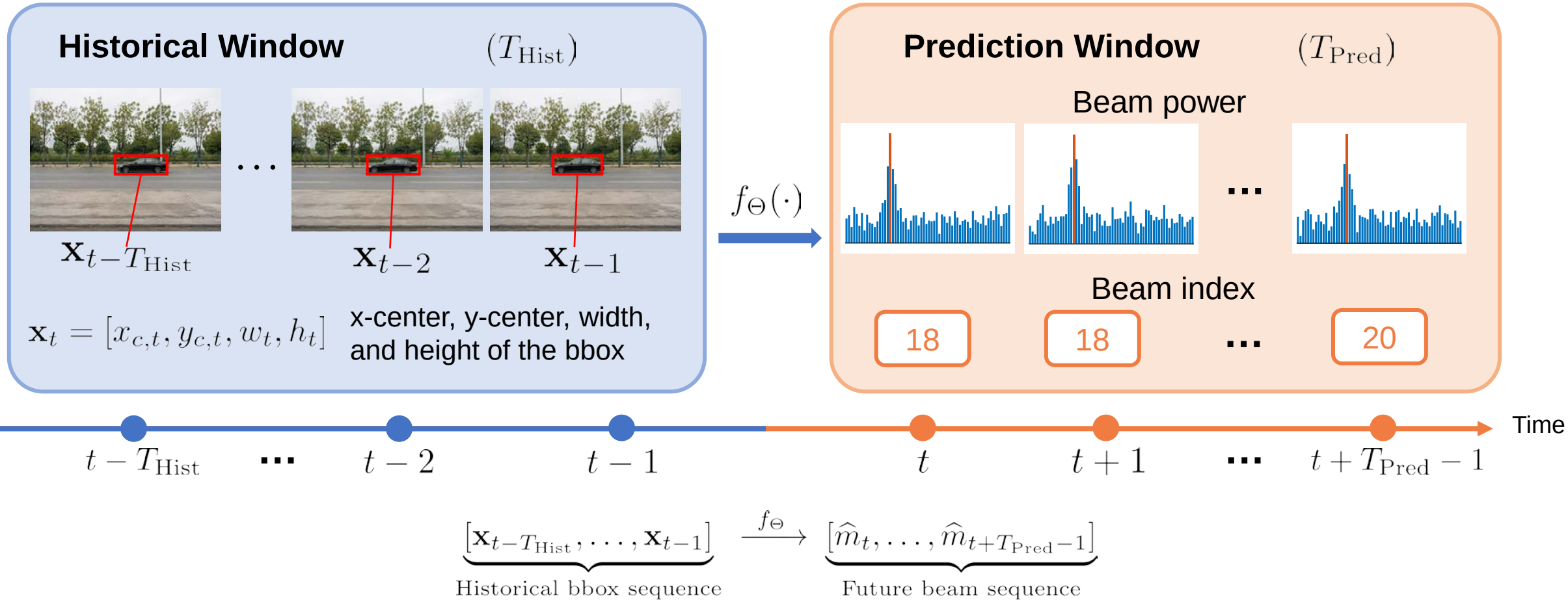
$$m_t^* = \arg \max_{m \in \{1,\ldots,M\}} \mathbf{p}_{t,m}$$

Select the beam with the highest power.

Power over 32 candidate beams



Vision-Aided Multi-Step Beam Prediction



During training

Bbox input + beam power / optimal-beam label

During inference

Historical bbox only.
Future information is not provided.

Contents

01 Background and Motivation

02 System Model and Problem Formulation

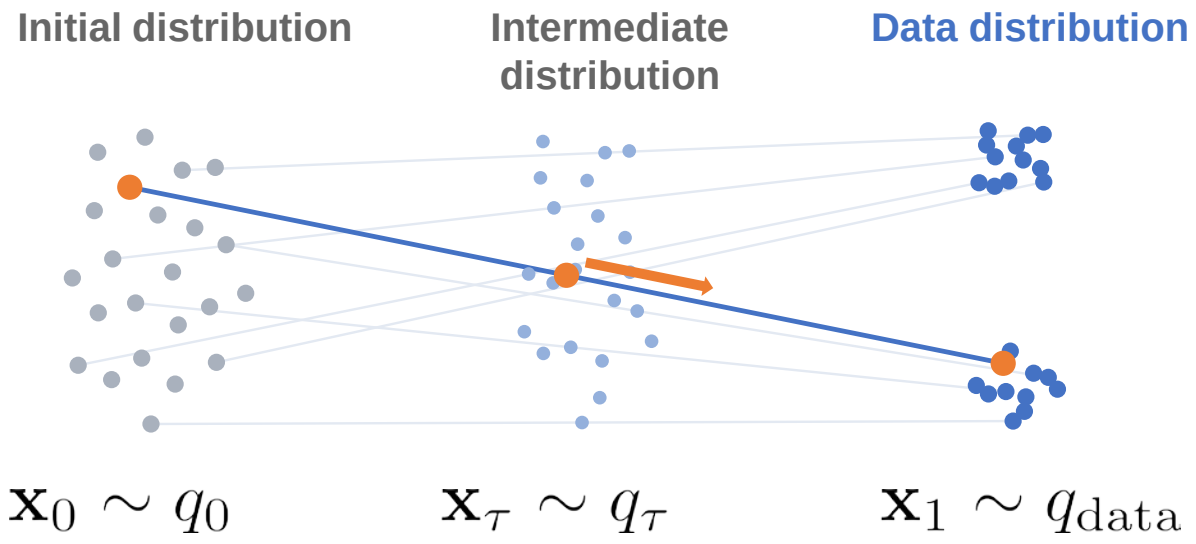
03 Proposed Method

04 Simulation Results

05 Conclusions

Flow Matching: Learning the Rule of Motion

A learned vector field transports an initial distribution toward the data distribution.



Generic FM illustration

This example starts from a source distribution.
Our predictor instead initializes from visual context.

1. Construct a local path

$$\mathbf{x}_\tau = (1-\tau)\mathbf{x}_0 + \tau\mathbf{x}_1,$$
$$\dot{\mathbf{x}}_\tau = \mathbf{x}_1 - \mathbf{x}_0.$$

Sample endpoint pairs and a uniform interpolation parameter.

2. Regress the analytic target

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{\tau,\mathbf{x}_0,\mathbf{x}_1}\left[\left\|\mathbf{u}_\theta(\mathbf{x}_\tau,\tau) - (\mathbf{x}_1 - \mathbf{x}_0)\right\|_2^2\right]$$

3. Integrate the learned field

$$\frac{d\hat{\mathbf{x}}_\tau}{d\tau} = \mathbf{u}_\theta(\hat{\mathbf{x}}_\tau,\tau), \qquad \hat{\mathbf{x}}_0 \sim q_0.$$

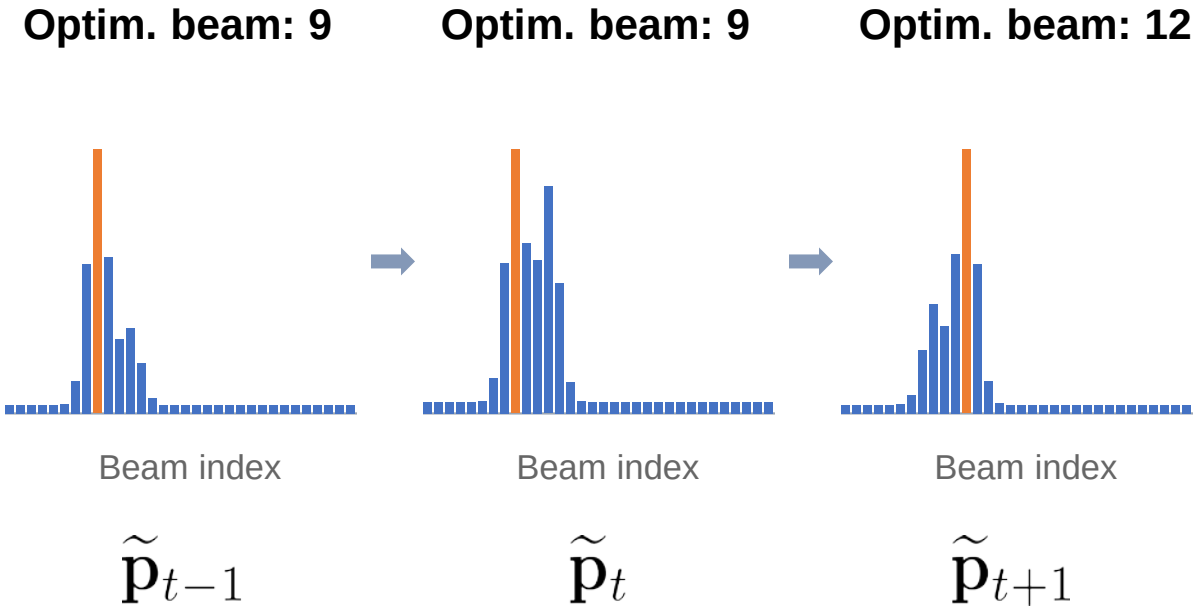
Train by regression; predict by integration.

Adapted from Holderrieth & Erives, MIT 6.S184 (2025), Lectures 2–3; linear-path example.

Why Flow Matching for Beam Prediction?

Learn local beam evolution from power supervision and observed vision history.

Power profiles reveal what a Optim. label hides



Why keep the full power vector?

The hard label keeps only the winning beam. The power vector also describes alternative candidates.

Why supervise adjacent frames?

Use neighboring power vectors to learn local beam evolution, rather than only connecting the window endpoints.

Segment-Wise Flow Matching Framework

1. Local FM supervision

$$k = t-1, \ldots, t+T_{\text{Pred}}-2, \quad \tau \sim \mathcal{U}(0,1)$$

$$\mathbf{z}_\tau = (1-\tau)\widetilde{\mathbf{p}}_k + \tau \widetilde{\mathbf{p}}_{k+1},$$

$$\dot{\mathbf{z}}_\tau = \widetilde{\mathbf{p}}_{k+1} - \widetilde{\mathbf{p}}_k.$$

$$\mathcal{L}_{\text{FM}} = \frac{1}{MT_{\text{Pred}}} \sum_{k=t-1}^{t+T_{\text{Pred}}-2} \mathbb{E}_\tau \left[\left\| \mathbf{u}_\theta(\mathbf{z}_\tau, \tau, \mathbf{c}) - \dot{\mathbf{z}}_\tau \right\|_2^2 \right]$$

2. Vision-conditioned prediction

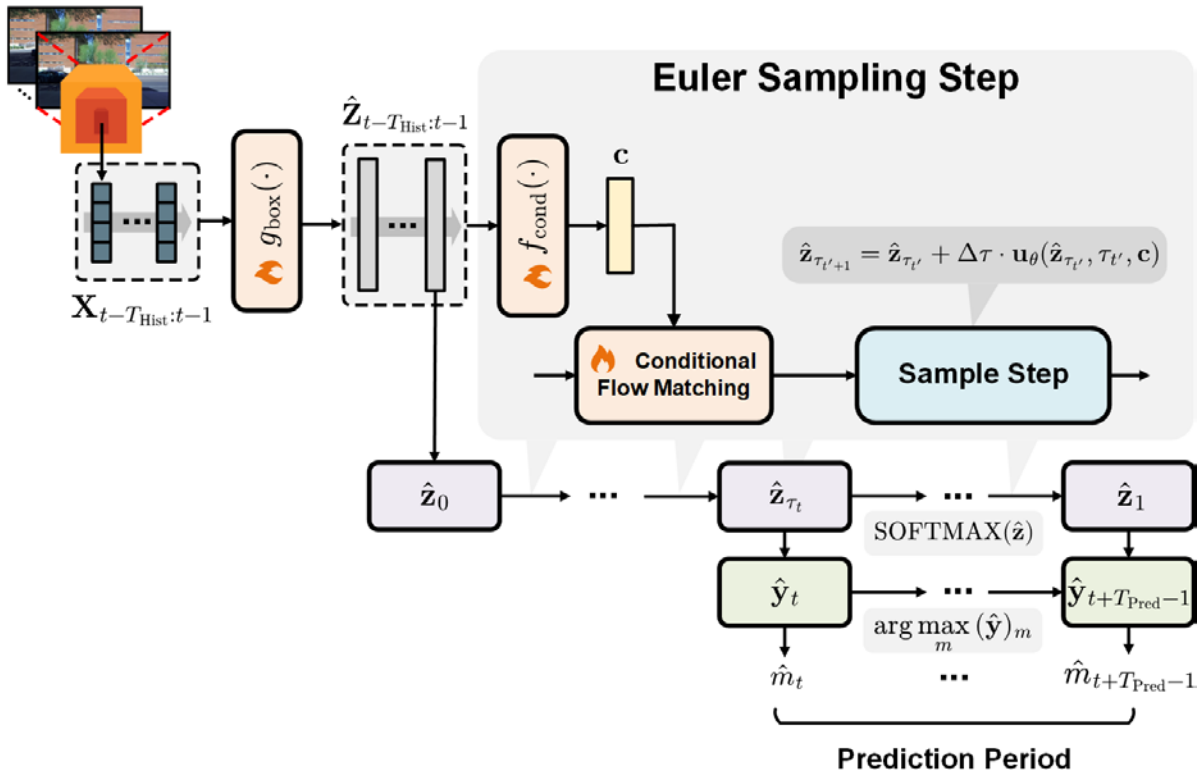
$$\mathbf{c} = f_{\text{cond}}(g_{\text{box}}(\mathbf{X}_{\text{Hist}})),$$

$$\widehat{\mathbf{z}}_0 = \text{Linear}(\mathbf{c}).$$

$$\widehat{\mathbf{z}}_{j+1} = \widehat{\mathbf{z}}_j + \frac{1}{T_{\text{Pred}}} \mathbf{u}_\theta(\widehat{\mathbf{z}}_j, \tau_j, \mathbf{c})$$

$$\tau_j = j / \max\{1, T_{\text{Pred}}-1\}, \quad j = 0, \ldots, T_{\text{Pred}}-1.$$

Architecture of the proposed framework



Training Objective

Classification

$$\mathcal{L}_{\text{CE}} = -\frac{1}{T_{\text{Pred}}} \sum_{t'=t}^{t+T_{\text{Pred}}-1} \log(\hat{\mathbf{y}}_{t',m^*_{t'}})$$

Flow Matching

$$\mathcal{L}_{\text{FM}}(\theta) = \mathrm{E}_{\tau}\left[\left\|\mathbf{u}_\theta(\mathbf{z}_\tau, \tau) - \frac{\tilde{\mathbf{p}}_{t'+1} - \tilde{\mathbf{p}}_{t'}}{\tau_{t'+1} - \tau_{t'}}\right\|^2\right]$$



Uncertainty Weighting Loss*

$$\mathcal{L} = \exp(-s_{\text{CE}})\,\mathcal{L}_{\text{CE}} + \exp(-s_{\text{FM}})\,\mathcal{L}_{\text{FM}} + \frac{1}{2}(s_{\text{CE}} + s_{\text{FM}})$$

$s_i = \log \sigma_i^2$

homoscedastic uncertainty of the task i

* A. Kendall, Y. Gal, and R. Cipolla, “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 7482–7491.

Contents

01 Background and Motivation

02 System Model and Problem Formulation

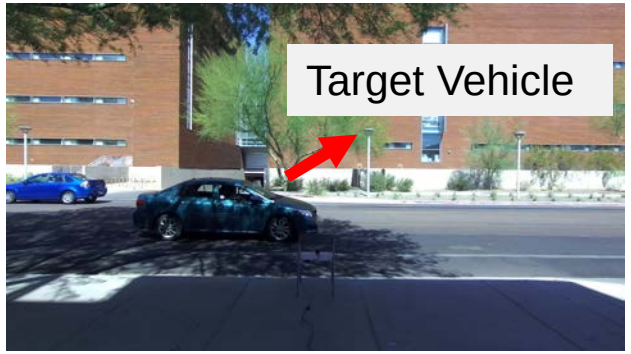
03 Proposed Method

04 Simulation Results

05 Conclusions

Simulation Settings

Dataset: DeepSense 6G



Scenario 8: Day-time

Prediction Configurations

Configuration A: $T_{\text{Hist}} = 8, T_{\text{Pred}} = 5$



Configuration B: $T_{\text{Hist}} = 3, T_{\text{Pred}} = 10$



Training Settings

Item	Setting
Codebook size	32
Batch size	64
Epochs	100
Optimizer	AdamW
Initial LR	0.001
LR schedule	10% warmup + 90% cosine decay
Weight decay	0.0001
Gradient clipping	1.0
Dropout	0.1

Evaluation Metrics: Top-K Accuracy ACC_K

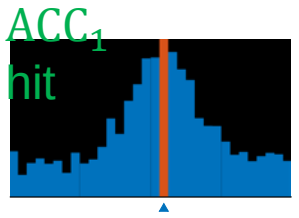
$$\text{ACC}_K = \frac{1}{N_{\text{Test}}} \sum_{n=1}^{N_{\text{Test}}} \mathbf{1}\left\{ m^{(n)*} \in \mathcal{Q}_K \right\}$$

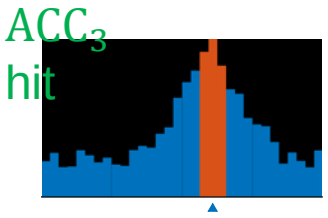
# of samples in the test set

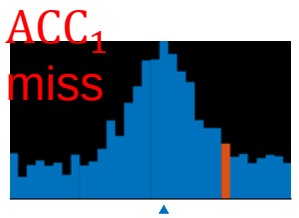
K beams with the highest predicted probabilities

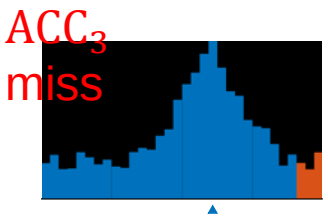
Metrics:

- Top-1 Accuracy ACC_1
- Top-3 Accuracy ACC_3





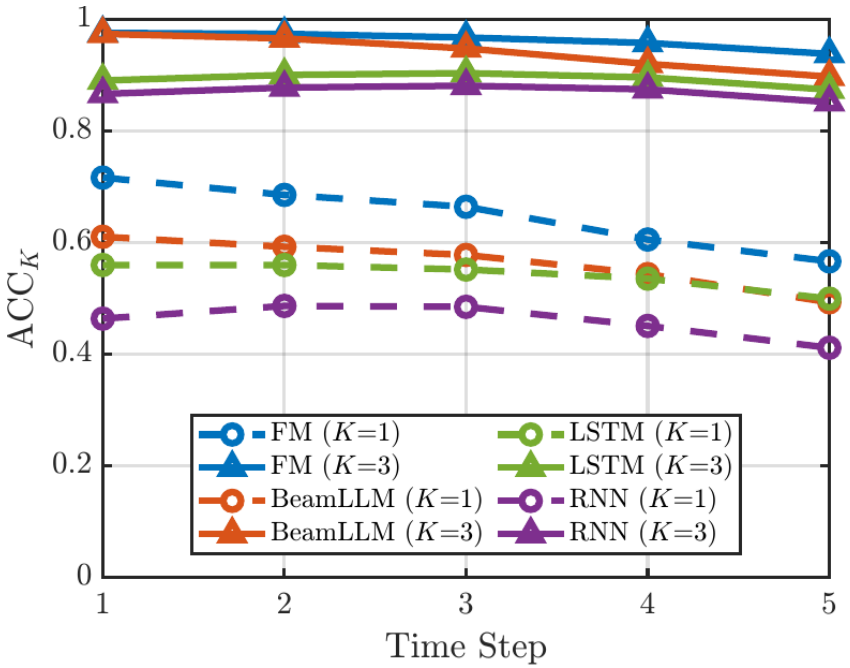




Simulation Results

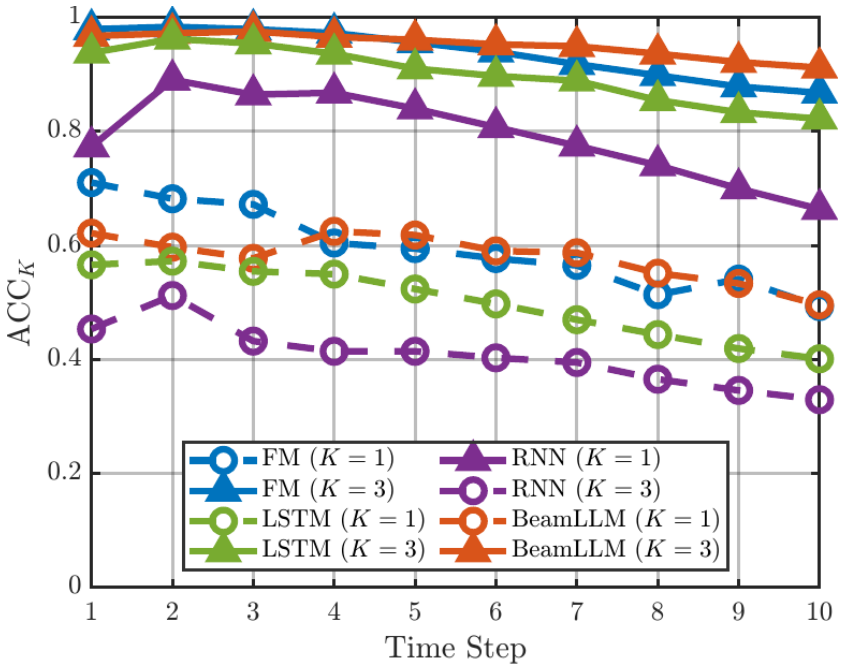
Config A: $T_{\text{Hist}} = 8, T_{\text{Pred}} = 5$

ACC_1/ ACC_3



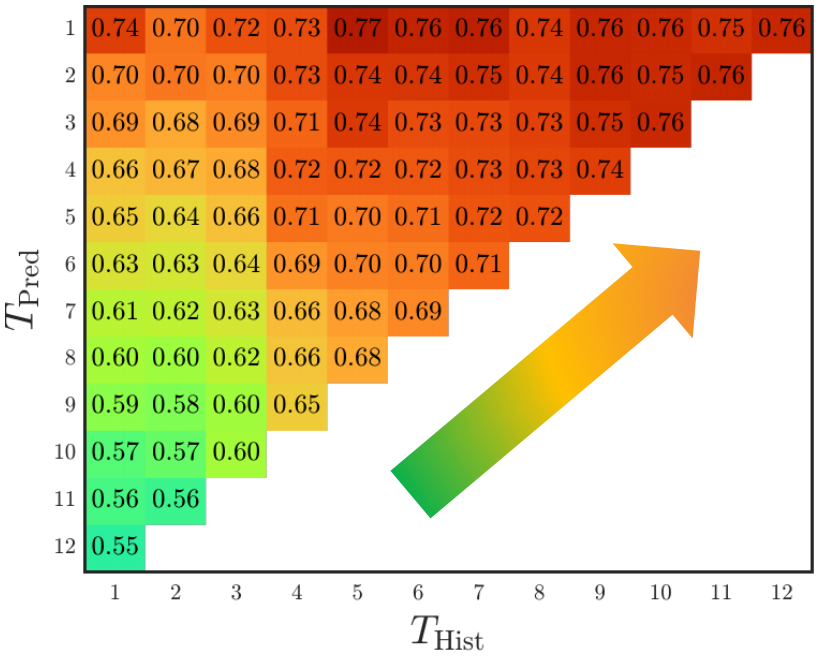
Config B: $T_{\text{Hist}} = 3, T_{\text{Pred}} = 10$

ACC_1/ ACC_3



Impact of T_{Hist} and T_{Pred}

ACC_1



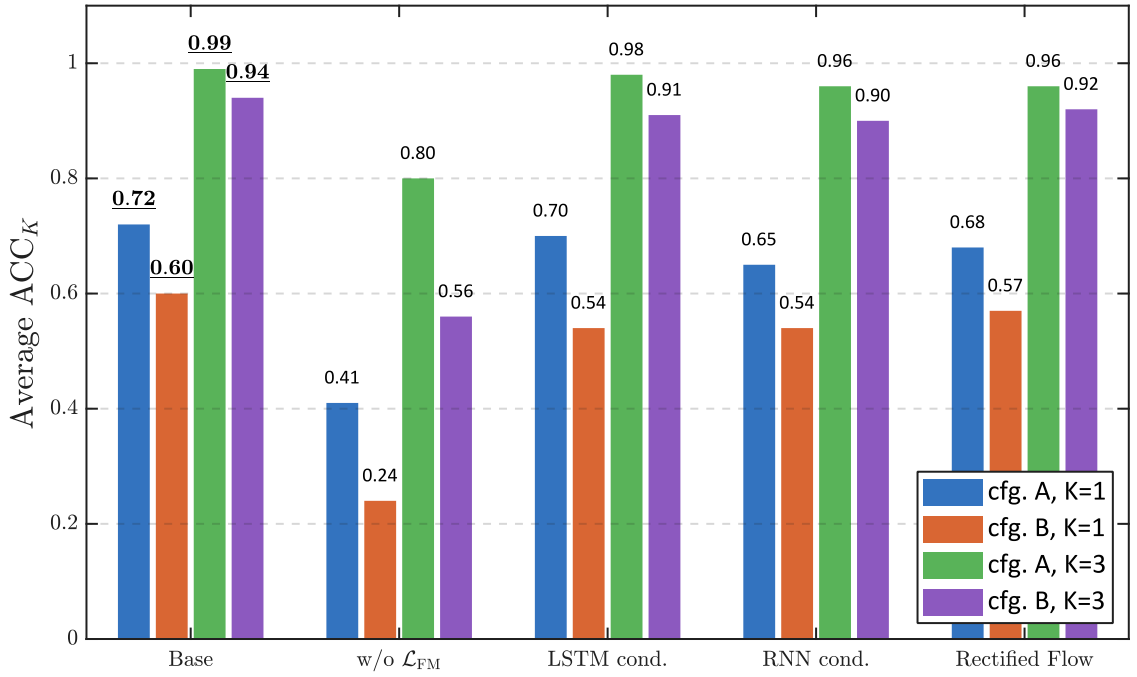
Short horizon: FM consistently achieves the highest accuracy on both ACC_1 and ACC_3.

Long horizon: FM gets higher accuracy in early steps; BeamLLM is more stable in later steps.

Increasing T_{Hist} generally improves accuracy; increasing T_{Pred} leads to accuracy degradation.

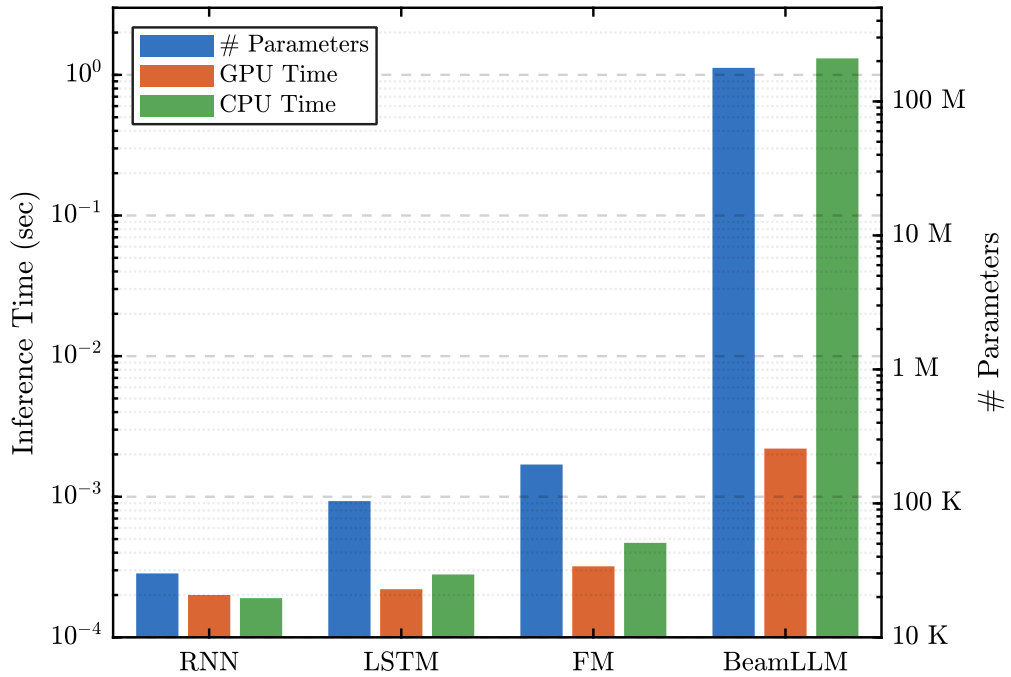
Ablation Studies and Complexity Analysis

Ablation Studies



- Base achieves the best accuracy across all settings.
- Removing $\mathcal{L}_{\mathrm{FM}}$ causes the largest drop.
- Condition encoder choice has limited impact.
- Segment-wise FM outperforms rectified flow.

Complexity Analysis



- FM offers the best accuracy–efficiency trade-off.
- BeamLLM has much higher parameter and latency cost.

Contents

01 Background and Motivation

02 System Model and Problem Formulation

03 Proposed Method

04 Simulation Results

05 Conclusions

Conclusions

- **Key Takeaways**
  - Segment-wise flow matching models **continuous** beam power evolution instead of discrete beam index sequences.
  - It outperforms RNN/LSTM and achieves competitive accuracy compared with BeamLLM.
  - Its **low inference cost** makes it suitable for edge deployment.
  - The flow matching loss is critical, especially for long-horizon prediction.
- **Future Works**
  - Extend to multi-user scenarios, diverse traffic densities, adverse weather, and different sensing configurations.
  - Apply flow matching to other tasks, such as proactive channel prediction and traffic/network load forecasting.

Thanks for your attention

* The visuals of the black sedan driving scene and the balance scale used in the slides were generated by ChatGPT.